Universidade Federal do Recôncavo da Bahia

[illegible]
[illegible]

[illegible]
[illegible]

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]

[illegible]
[illegible]
[illegible]

[illegible]
[illegible]
[illegible]

⠨⠉⠊⠇⠁ ⠙⠑ ⠏⠊⠁ ⠠⠉⠁⠎⠞⠗⠕
⠠⠁⠙⠗⠊⠁⠝⠁ ⠠⠉⠁⠗⠙⠕⠎⠕

⠠⠓⠊⠎⠞⠕⠗⠊⠁⠎ ⠙⠑ ⠍⠁⠗⠉⠁⠎ ⠙⠁ ⠓⠊⠎⠞⠕⠗⠊⠁ ⠙⠁ ⠁⠋⠗⠁⠝⠞⠁⠍⠑⠝⠞⠕

Editora UFRB

Universidade Federal do Recôncavo da Bahia

Reitor: Prof. Dr. Paulo Roberto Pereira de Santana
Vice-Reitora: Profa. Dra. Georgina Gonçalves dos Santos

Superintendência de Gestão Administrativa
Superintendente: Prof. Dr. Fábio Fernandes Pinto

Superintendência de Infraestrutura
Superintendente: Prof. Dr. Ricardo Alexandre Amaral de Moraes

Superintendência de Gestão Acadêmica
Superintendente: Profa. Dra. Ana Paula Oliveira Marques

Superintendência de Ações Afirmativas, Diversidade e Inclusão
Superintendente: Profa. Dra. Cristiane Maria Amorim Rocha

Superintendência de Tecnologia da Informação
Superintendente: Prof. Dr. Fabrício Vieira da Silva

Superintendência de Relações Internacionais
Superintendente: Prof. Dr. Marcelo Henrique Otenio

Superintendência de Extensão
Superintendente: Prof. Dr. Ricardo Queiroz Pinheiro

Superintendência de Pós-Graduação, Pesquisa e Inovação
Superintendente: Prof. Dr. Marcelo Schramm Mattos

Conselho Editorial da Editora UFRB
Presidente: Prof. Dr. Antonio Fernando Guerreiro de Freitas Penteado

Membros Titulares
Profa. Dra. Adriana Nunes de Miranda Zanoni
Profa. Dra. Ana Rita Coimbra Mota
Profa. Dra. Ana Rosa Peixoto Brito
Profa. Dra. Cristiane Maria Amorim Rocha
Profa. Dra. Cristina Maria Rosa
Profa. Dra. Daniela Nascimento Arbex
Profa. Dra. Elisabete Cristina Sobreira Lima
Profa. Dra. Georgina Gonçalves dos Santos
Prof. Dr. Jorge Gustavo Velame Fialho Feitosa
Prof. Dr. José Roberto Severino
Profa. Dra. Lícia Maria Freire
Profa. Dra. Lívia Diana Rocha Magalhães
Profa. Dra. Maria Fernanda Salcedo Repolês Maluf
Profa. Dra. Maria Lúcia Pereira Souza Gabriel
Profa. Dra. Mariluce Souza Pereira
Profa. Dra. Renata Prado Soares
Prof. Dr. Ricardo Alexandre Amaral de Moraes
Profa. Dra. Rosângela Pereira de Tugny
Profa. Dra. Tatiane Cosentino Rodrigues
Profa. Dra. Vanessa Faria Lima

Membros Suplentes
Profa. Dra. Ana Paula Oliveira Marques
Prof. Dr. Antonio Fernando Guerreiro de Freitas Penteado
Profa. Dra. Cristiane Maria Amorim Rocha
Profa. Dra. Daniela Nascimento Arbex
Profa. Dra. Elisabete Cristina Sobreira Lima
Profa. Dra. Georgina Gonçalves dos Santos
Profa. Dra. Lícia Maria Freire
Profa. Dra. Lívia Diana Rocha Magalhães
Profa. Dra. Maria Fernanda Salcedo Repolês Maluf
Profa. Dra. Mariluce Souza Pereira
Profa. Dra. Renata Prado Soares
Profa. Dra. Rosângela Pereira de Tugny
Profa. Dra. Tatiane Cosentino Rodrigues
Profa. Dra. Vanessa Faria Lima

Editora UFRB

Rua Rui Barbosa, 710 - Centro
Cruz das Almas - Bahia
CEP 44.380-000
Fone: (75) 3621-6550
www.ufrb.edu.br/editora
editora@ufrb.edu.br

⠠⠁⠄ ⠠⠁⠄ ⠄⠄⠠⠄⠁⠄⠄ ⠄ ⠠⠄⠁ ⠄ ⠄⠄⠁⠠⠄⠠ ⠄⠄
⠄⠁⠄⠁⠄⠁⠁ ⠄⠠⠁⠁ ⠄⠄⠁ ⠄⠄⠠⠄⠁ ⠄ ⠄⠄ ⠠⠄⠄⠠
⠄ ⠠⠁⠄⠁⠠⠄⠄⠁ ⠄⠁⠠⠄⠁ ⠄⠄⠁⠄⠁⠠⠄⠁⠠⠄ ⠄⠁⠠
⠄ ⠄⠄⠁⠄⠠ ⠁⠁⠁⠄⠁⠠⠄ ⠄⠄⠁⠄⠠⠁⠠⠄⠁⠄⠁⠠⠄
⠄ ⠁⠁⠁⠄⠄⠄ ⠄⠄⠁⠄⠄⠄⠁⠠⠄ ⠄⠄⠄⠁⠄⠁⠠⠄⠁⠄⠁⠠⠄

⠠⠠⠠

⠠⠓⠕⠍⠑

⠠⠞⠓⠑⠕⠗⠊⠑⠎ ⠕⠋ ⠠⠍⠕⠞⠊⠧⠁⠞⠊⠕⠝ ..... ⠼⠁
⠀⠀⠀

⠠⠞⠓⠑ ⠠⠍⠁⠝ ⠺⠓⠕ ⠺⠁⠎ ⠞⠕⠕
⠒⠒⠒

⠠⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠕⠝ ⠠⠠⠠

⠠⠞⠓⠑ ⠠⠎⠁⠝⠙

⠠⠉⠓⠁⠏⠞⠑⠗ ⠼⠁⠚

⠀⠀⠀⠀⠠⠞⠓⠑ ⠠⠎⠁⠝⠙ ⠠⠺⠁⠎ ⠁ ⠠⠏⠗⠑⠞⠞⠽ ⠇⠁⠗⠛⠑ ⠞⠗⠑⠑ ⠞⠓⠁⠞
⠠⠺⠁⠎ ⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠞⠕ ⠠⠙⠕⠗⠁⠎ ⠠⠃⠁⠗⠗⠽⠍⠕⠗⠑ ⠠⠃⠁⠗⠗⠽⠍⠕⠗⠑ ⠠⠺⠁⠎
⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠞⠕ ⠠⠙⠕⠗⠁⠎ ⠠⠃⠁⠗⠗⠽⠍⠕⠗⠑ ⠠⠺⠁⠎ ⠁ ⠠⠏⠗⠑⠞⠞⠽ ⠇⠁⠗⠛⠑
⠞⠗⠑⠑ ⠞⠓⠁⠞ ⠠⠺⠁⠎ ⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠞⠕ ⠠⠙⠕⠗⠁⠎ ⠠⠃⠁⠗⠗⠽⠍⠕⠗⠑ ⠠⠺⠁⠎
⠁ ⠠⠏⠗⠑⠞⠞⠽ ⠇⠁⠗⠛⠑ ⠞⠗⠑⠑ ⠞⠓⠁⠞ ⠠⠺⠁⠎ ⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠞⠕ ⠠⠙⠕⠗⠁⠎
⠠⠃⠁⠗⠗⠽⠍⠕⠗⠑ ⠠⠺⠁⠎ ⠁ ⠠⠏⠗⠑⠞⠞⠽ ⠇⠁⠗⠛⠑ ⠞⠗⠑⠑ ⠞⠓⠁⠞ ⠠⠺⠁⠎ ⠙⠑⠙⠊⠉⠁⠞⠑⠙
⠞⠕ ⠠⠙⠕⠗⠁⠎ ⠠⠃⠁⠗⠗⠽⠍⠕⠗⠑ ⠠⠺⠁⠎ ⠁ ⠠⠏⠗⠑⠞⠞⠽ ⠇⠁⠗⠛⠑ ⠞⠗⠑⠑ ⠞⠓⠁⠞
⠠⠺⠁⠎ ⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠞⠕ ⠠⠙⠕⠗⠁⠎ ⠠⠃⠁⠗⠗⠽⠍⠕⠗⠑ ⠠⠺⠁⠎ ⠁ ⠠⠏⠗⠑⠞⠞⠽
⠇⠁⠗⠛⠑ ⠞⠗⠑⠑ ⠞⠓⠁⠞ ⠠⠺⠁⠎ ⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠞⠕ ⠠⠙⠕⠗⠁⠎ ⠠⠃⠁⠗⠗⠽⠍⠕⠗⠑
⠠⠺⠁⠎

⠠⠁ ⠠⠗⠑⠏⠗⠑⠎⠑⠝⠞⠁⠞⠊⠧⠑ ⠙⠑⠍⠕⠉⠗⠁⠉⠽ ⠊⠎ ⠁ ⠋⠕⠗⠍ ⠕⠋ ⠛⠕⠧⠑⠗⠝⠍⠑⠝⠞ ⠺⠓⠑⠗⠑ ⠞⠓⠑ ⠏⠑⠕⠏⠇⠑ ⠺⠓⠕ ⠁⠗⠑ ⠛⠕⠧⠑⠗⠝⠑⠙ ⠁⠝⠙ ⠞⠓⠑ ⠞⠗⠁⠝⠎⠁⠉⠞⠊⠕⠝⠎ ⠁⠗⠑ ⠉⠕⠝⠙⠥⠉⠞⠑⠙ ⠁⠉⠉⠕⠗⠙⠊⠝⠛ ⠞⠕ ⠞⠓⠑ ⠗⠥⠇⠑⠎ ⠕⠋ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠞⠓⠑ ⠁⠗⠞⠎ ⠁⠝⠙ ⠞⠓⠑ ⠉⠕⠍⠍⠕⠝⠺⠑⠁⠇⠞⠓ ⠕⠋ ⠍⠁⠎⠎⠁⠉⠓⠥⠎⠑⠞⠞⠎ ⠁⠝⠙ ⠞⠓⠑ ⠥⠝⠊⠧⠑⠗⠎⠁⠇ ⠙⠑⠉⠇⠁⠗⠁⠞⠊⠕⠝ ⠕⠋ ⠓⠥⠍⠁⠝ ⠗⠊⠛⠓⠞⠎ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠍⠊⠎⠑ ⠕⠋ ⠁ ⠙⠑⠍⠕⠉⠗⠁⠞⠊⠉ ⠎⠕⠉⠊⠑⠞⠽ ⠁⠝⠙ ⠞⠓⠑ ⠗⠥⠇⠑ ⠕⠋ ⠇⠁⠺ ⠊⠝ ⠞⠓⠑ ⠏⠗⠕⠞⠑⠉

Machine Learning Applications

Machine learning has revolutionized many fields, from computer vision to natural language processing. In this chapter, we explore advanced techniques and their applications.

Deep Neural Networks

Deep neural networks consist of multiple layers of interconnected nodes. Each layer transforms the input data, allowing the network to learn complex patterns.

Convolutional Neural Networks

Convolutional neural networks (CNNs) are particularly effective for image recognition tasks. They use convolution operations to detect features such as edges, textures, and shapes.

Recurrent Neural Networks

Recurrent neural networks (RNNs) are designed for sequential data. They maintain a hidden state that captures information about previous inputs, making them suitable for time series and language modeling.

Reinforcement Learning

Reinforcement learning involves training an agent to make decisions by rewarding desirable actions and penalizing undesirable ones. The agent learns a policy that maximizes the cumulative reward over time.

Q-Learning

Q-learning is a model-free reinforcement learning algorithm. It learns the value of taking a particular action in a particular state, without requiring a model of the environment.

Policy Gradient Methods

Policy gradient methods directly optimize the policy function. They are particularly useful for problems with continuous action spaces.

Unsupervised Learning

Unsupervised learning discovers hidden patterns in unlabeled data. Common techniques include clustering and dimensionality reduction.

Clustering

Clustering groups similar data points together. K-means and hierarchical clustering are two popular algorithms.

Dimensionality Reduction

Dimensionality reduction techniques like principal component analysis (PCA) reduce the number of features while preserving the most important information.

Evaluation Metrics

Evaluating machine learning models requires appropriate metrics. Accuracy, precision, recall, and F1 score are commonly used for classification tasks.

Cross-Validation

Cross-validation helps assess how well a model generalizes to unseen data. It involves splitting the data into multiple folds and training on different combinations.

Conclusion

Machine learning continues to advance rapidly. Understanding the fundamentals and staying current with new developments is essential for practitioners.

⠠⠁⠇⠇ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠊⠑⠎ ⠁⠝⠙ ⠏⠑⠗⠎⠕⠝⠎ ⠊⠝ ⠞⠓⠑ ⠏⠗⠑⠉⠑⠙⠊⠝⠛
⠞⠓⠊⠎ ⠉⠕⠝⠧⠑⠝⠞⠊⠕⠝ ⠁⠗⠑ ⠋⠊⠉⠞⠊⠞⠊⠕⠥⠎ ⠁⠝⠙ ⠁⠝⠽
⠎⠊⠍⠊⠇⠁⠗⠊⠞⠽ ⠊⠎ ⠉⠕⠊⠝⠉⠊⠙⠑⠝⠞⠁⠇⠲ ⠠⠁ ⠉⠗⠑⠙⠊⠞ ⠞⠕
⠞⠓⠑ ⠁⠉⠞⠥⠁⠇ ⠏⠑⠗⠎⠕⠝ ⠕⠗ ⠉⠕⠍⠏⠁⠝⠽ ⠊⠎ ⠝⠕⠞
⠝⠑⠉⠑⠎⠎⠁⠗⠽ ⠁⠝⠙ ⠁⠝⠽ ⠎⠊⠍⠊⠇⠁⠗⠊⠞⠽ ⠊⠎
⠉⠕⠊⠝⠉⠊⠙⠑⠝⠞⠁⠇⠲

⠠⠞⠓⠊⠎ ⠏⠁⠛⠑ ⠺⠁⠎ ⠉⠗⠑⠁⠞⠑⠙ ⠊⠝ ⠼⠁⠚⠚⠚

⠠⠓⠊⠎⠞⠕⠗⠽ ⠕⠋ ⠠⠁⠗⠞ ⠊⠝ ⠠⠁⠍⠑⠗⠊⠉⠁
⠠⠉⠕⠇⠕⠝⠊⠁⠇ ⠠⠁⠗⠉⠓⠊⠞⠑⠉⠞⠥⠗⠑

⠠⠃⠽ ⠠⠍⠁⠗⠊⠁ ⠠⠎⠁⠝⠞⠁ ⠠⠉⠗⠥⠵

⠠⠃⠽ ⠠⠎⠁⠍⠥⠑⠇ ⠠⠑⠇⠊⠕⠞ ⠠⠍⠕⠗⠝⠑⠞

⠠⠁ ⠠⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑ ⠠⠓⠊⠎⠞⠕⠗⠽ ⠕⠋ ⠠⠁⠗⠉⠓⠊⠞⠑⠉⠞⠥⠗⠑, ⠋⠗⠕⠍ ⠠⠍⠑⠎⠕⠏⠕⠞⠁⠍⠊⠁ ⠞⠕ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠞⠓⠗⠕⠥⠛⠓ ⠞⠓⠑ ⠠⠉⠕⠇⠕⠝⠊⠁⠇ ⠠⠑⠗⠁, ⠞⠓⠑ ⠠⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞ ⠠⠗⠑⠏⠥⠃⠇⠊⠉ ⠕⠋ ⠠⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎, ⠞⠓⠑ ⠠⠉⠕⠍⠍⠕⠝⠺⠑⠁⠇⠞⠓ ⠕⠋ ⠠⠍⠁⠎⠎⠁⠉⠓⠥⠎⠑⠞⠞⠎, ⠁⠝⠙ ⠞⠓⠑ ⠠⠙⠊⠎⠞⠗⠊⠉⠞ ⠕⠋ ⠠⠉⠕⠇⠥⠍⠃⠊⠁, ⠁⠝⠙ ⠞⠕ ⠠⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠑⠗⠗⠊⠞⠕⠗⠊⠑⠎ ⠠⠁⠝⠙ ⠠⠙⠑⠏⠑⠝⠙⠑⠝⠉⠊⠑⠎ ⠠⠁⠝⠙ ⠠⠏⠕⠎⠎⠑⠎⠎⠊⠕⠝⠎ ⠠⠁⠗⠕⠥⠝⠙ ⠞⠓⠑ ⠠⠠⠺⠕⠗⠇⠙.

⠠⠁⠝⠙ ⠠⠺⠓⠑⠗⠑⠁⠎ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠠⠕⠝ ⠠⠁⠗⠉⠓⠊⠧⠑⠎ ⠠⠁⠝⠙ ⠠⠍⠁⠝⠥⠎⠉⠗⠊⠏⠞⠎ ⠠⠁⠝⠙ ⠠⠕⠞⠓⠑⠗ ⠠⠙⠕⠉⠥⠍⠑⠝⠞⠎ ⠠⠕⠋ ⠠⠓⠊⠎⠞⠕⠗⠽ ⠠⠁⠗⠑ ⠠⠎⠕ ⠠⠉⠕⠝⠙⠥⠉⠞⠑⠙ ⠠⠁⠎ ⠠⠞⠕ ⠠⠍⠁⠅⠑ ⠠⠊⠞ ⠠⠏⠕⠎⠎⠊⠃⠇⠑ ⠠⠋⠕⠗ ⠠⠁⠝⠽ ⠠⠏⠑⠗⠎⠕⠝ ⠠⠞⠕ ⠠⠎⠑⠁⠗⠉⠓ ⠠⠋⠕⠗ ⠠⠞⠗⠥⠞⠓ ⠠⠁⠝⠙ ⠠⠙⠊⠎⠉⠕⠧⠑⠗ ⠠⠞⠗⠥⠞⠓ ⠠⠁⠝⠙ ⠠⠞⠗⠥⠞⠓ ⠠⠕⠝⠇⠽.

⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠎⠑⠉⠕⠝⠙ ⠠⠞⠓⠁⠞ ⠠⠞⠓⠑ ⠠⠎⠁⠍⠑ ⠠⠁⠎ ⠠⠞⠓⠑ ⠠⠋⠊⠗⠎⠞, ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠞⠓⠊⠗⠙ ⠠⠞⠓⠁⠞ ⠠⠞⠓⠑ ⠠⠎⠁⠍⠑ ⠠⠁⠎ ⠠⠞⠓⠑ ⠠⠎⠑⠉⠕⠝⠙, ⠠⠁⠗⠑ ⠠⠞⠓⠑ ⠠⠞⠗⠥⠑ ⠠⠁⠝⠙ ⠠⠉⠕⠍⠏⠇⠑⠞⠑ ⠠⠠⠺⠕⠗⠇⠙ ⠠⠕⠋ ⠠⠁⠗⠉⠓⠊⠞⠑⠉⠞⠥⠗⠑

⠠⠁⠃⠎⠞⠗⠁⠉⠞⠒ ⠠⠊⠝ ⠞⠓⠊⠎ ⠏⠁⠏⠑⠗⠂ ⠺⠑ ⠙⠑⠎⠉⠗⠊⠃⠑ ⠞⠓⠑ ⠙⠑⠧⠑⠇⠕⠏⠍⠑⠝⠞ ⠕⠋ ⠁
⠝⠑⠺ ⠞⠑⠉⠓⠝⠊⠟⠥⠑ ⠞⠕ ⠉⠕⠍⠏⠥⠞⠑ ⠁ ⠝⠥⠍⠃⠑⠗ ⠕⠋ ⠏⠕⠎⠎⠊⠃⠇⠑ ⠎⠕⠇⠥⠞⠊⠕⠝⠎ ⠋⠕⠗
⠁ ⠛⠊⠧⠑⠝ ⠏⠗⠕⠃⠇⠑⠍⠂ ⠁ ⠉⠑⠗⠞⠁⠊⠝ ⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠍⠑⠞⠓⠕⠙ ⠊⠎ ⠝⠑⠉⠑⠎⠎⠁⠗⠽
⠞⠕ ⠙⠑⠞⠑⠗⠍⠊⠝⠑ ⠁ ⠎⠕⠇⠥⠞⠊⠕⠝ ⠞⠕ ⠁ ⠏⠗⠕⠃⠇⠑⠍ ⠞⠓⠁⠞ ⠊⠎ ⠝⠕⠞ ⠎⠕⠇⠧⠁⠃⠇⠑ ⠊⠝
⠁ ⠉⠇⠕⠎⠑⠙ ⠎⠽⠎⠞⠑⠍⠂ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠉⠑⠎⠎ ⠕⠋ ⠙⠑⠞⠑⠗⠍⠊⠝⠊⠝⠛ ⠺⠓⠁⠞
⠊⠎ ⠉⠁⠇⠉⠥⠇⠁⠃⠇⠑ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠉⠑⠙⠥⠗⠑⠎ ⠞⠓⠁⠞ ⠍⠥⠎⠞ ⠃⠑ ⠋⠕⠇⠇⠕⠺⠑⠙
⠞⠕ ⠉⠕⠍⠏⠥⠞⠑ ⠁ ⠎⠕⠇⠥⠞⠊⠕⠝⠲

⠠⠠⠊⠍⠏⠕⠗⠞⠁⠝⠉⠑ ⠕⠋ ⠁⠇⠛⠕⠗⠊⠞⠓⠍⠎

⠠⠊⠝ ⠞⠓⠑ ⠋⠊⠗⠎⠞ ⠉⠓⠁⠏⠞⠑⠗⠂ ⠺⠑ ⠙⠑⠋⠊⠝⠑ ⠁⠇⠛⠕⠗⠊⠞⠓⠍⠎ ⠁⠝⠙
⠙⠊⠎⠉⠥⠎⠎ ⠞⠓⠑⠊⠗ ⠊⠍⠏⠕⠗⠞⠁⠝⠉⠑ ⠊⠝ ⠞⠓⠑ ⠉⠕⠝⠞⠑⠭⠞ ⠕⠋ ⠞⠓⠑ ⠏⠗⠕⠃⠇⠑⠍⠎
⠺⠑ ⠁⠙⠙⠗⠑⠎⠎⠲ ⠠⠊⠞ ⠊⠎ ⠊⠍⠏⠕⠗⠞⠁⠝⠞ ⠞⠕ ⠝⠕⠞⠑ ⠞⠓⠁⠞ ⠞⠓⠑ ⠏⠗⠕⠃⠇⠑⠍⠎ ⠺⠑
⠁⠙⠙⠗⠑⠎⠎ ⠁⠗⠑ ⠝⠕⠞ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠺⠑ ⠁⠗⠑ ⠎⠕⠇⠧⠊⠝⠛⠲ ⠠⠞⠓⠑
⠏⠗⠕⠃⠇⠑⠍⠎ ⠁⠗⠑ ⠙⠊⠋⠋⠑⠗⠑⠝⠞⠂ ⠁⠝⠙ ⠞⠓⠑⠗⠑⠋⠕⠗⠑ ⠞⠓⠑ ⠎⠕⠇⠥⠞⠊⠕⠝⠎ ⠁⠗⠑
⠙⠊⠋⠋⠑⠗⠑⠝⠞⠲

⠠⠠⠠

⠇ ⠍⠝⠑ ⠞⠓ ⠞⠗⠁⠙⠑ ⠞⠗⠑⠁⠞⠊⠑⠎ ⠞⠓⠁⠞ ⠃⠑⠛⠁⠝ ⠺⠊⠞⠓ ⠎⠍⠁⠇⠇ ⠏⠗⠊⠝⠞⠎ ⠕⠋ ⠍⠝⠑ ⠞⠓
⠎ ⠞⠓⠗⠑⠑ ⠞⠽⠏⠑⠎ ⠕⠋ ⠏⠗⠊⠝⠞⠎ ⠁⠇⠇ ⠺⠑⠗⠑ ⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠋⠗⠕⠍ ⠞⠓
⠕⠞ ⠽⠑⠞ ⠺⠑ ⠍⠁⠙⠑ ⠞⠓⠑ ⠞⠗⠁⠙⠑ ⠊⠞ ⠺⠁⠎ ⠃⠑⠛⠁⠝ ⠺⠊⠞⠓ ⠍⠝⠑
⠞⠓⠗⠑⠑ ⠍⠁⠝⠥⠁⠇ ⠍⠝⠑⠍⠕⠝⠊⠉⠎ ⠁ ⠋⠑⠺ ⠍⠝⠑ ⠁⠙⠙⠑⠙ ⠁ ⠋⠑⠺ ⠍⠕⠗⠑
⠇ ⠍⠝⠑ ⠺⠁⠎ ⠍⠁⠙⠑ ⠍⠝⠑⠍⠕⠝⠊⠉⠎ ⠕⠝ ⠞⠓⠗⠑⠑⠞⠽⠏⠑⠎ ⁂ ⠁ ⠋⠑⠺
⠞⠓⠗⠑⠑⠞⠽⠏⠑⠎ ⁂ ⠊ ⠃⠑⠛⠁⠝ ⠺⠊⠞⠓

⠠ ⠠⠠ ⠃⠥⠞ ⠊ ⠓⠁⠙ ⠝⠕⠞ ⠙⠕⠝⠑ ⠁⠝⠽ ⠺⠕⠗⠅ ⠁⠞ ⠁⠇⠇ ⠁⠝⠙ ⠺⠁⠎ ⠙⠑⠎⠏⠁⠊⠗⠊⠝⠛
⠏⠗⠊⠝⠞⠎ ⠁⠗⠑ ⠁⠇⠗⠑⠁⠙⠽ ⠃⠑⠞⠞⠑⠗ ⠞⠓⠁⠝ ⠞⠓⠑⠽ ⠁⠗⠑ ⠞⠕⠙⠁⠽ ⠠⠠ ⠺⠑ ⠍⠁⠽ ⠎⠞⠊⠇⠇ ⠓⠁⠧⠑
⠍⠥⠉⠓ ⠺⠕⠗⠅ ⠞⠕ ⠙⠕ ⠠⠠ ⠺⠑ ⠍⠁⠽ ⠁⠇⠎⠕ ⠓⠁⠧⠑ ⠍⠁⠝⠽ ⠏⠗⠊⠝⠞⠎ ⠞⠓⠁⠞ ⠁⠗⠑ ⠙⠊⠋⠋⠑⠗⠑⠝⠞
⠠ ⠠⠠ ⠞⠓⠊⠝⠅ ⠞⠓⠁⠞ ⠺⠑ ⠙⠕ ⠝⠕⠞ ⠓⠁⠧⠑ ⠁ ⠍⠁⠝⠥⠁⠇ ⠕⠋ ⠁⠇⠇ ⠞⠓⠑ ⠏⠗⠊⠝⠞⠎ ⠺⠑
⠓⠁⠧⠑ ⠁⠝⠙ ⠞⠓⠁⠞ ⠊⠎ ⠺⠓⠁⠞ ⠍⠁⠅⠑⠎ ⠞⠓⠊⠎ ⠎⠕ ⠙⠊⠋⠋⠊⠉⠥⠇⠞ ⠠⠠ ⠞⠓⠊⠝⠅
⠠ ⠠⠠ ⠞⠓⠊⠝⠅ ⠠⠠ ⠍⠁⠽ ⠓⠁⠧⠑ ⠁ ⠍⠁⠝⠥⠁⠇ ⠕⠋ ⠁⠇⠇ ⠞⠓⠑ ⠏⠗⠊⠝⠞⠎ ⠺⠑ ⠓⠁⠧⠑

⠠⠞⠓⠑ ⠠⠁⠝⠝⠁⠇⠎ ⠕⠋ ⠠⠞⠓⠑ ⠠⠺⠕⠗⠇⠙ ⠠⠞⠗⠁⠙⠑ ⠠⠕⠗⠛⠁⠝⠊⠵⠁⠞⠊⠕⠝ ⠠⠞⠓⠗⠕⠥⠛⠓ ⠠⠞⠓⠑ ⠠⠁⠛⠑ ⠕⠋ ⠠⠁⠝⠁⠗⠉⠓⠽ ⠼⠃

⠀⠀⠀

⠠⠠⠠

Introduction to Quantum Computing

Chapter 1

Quantum computing is a rapidly evolving field that leverages the principles of quantum mechanics to perform computations. Unlike classical computers, which use bits to represent information as either 0 or 1, quantum computers use quantum bits, or qubits, which can exist in a superposition of both states simultaneously. This property allows quantum computers to process a vast number of possibilities at once.

Qubits and Superposition

A qubit can be represented as a linear combination of the basis states $|0\rangle$ and $|1\rangle$. The general state of a qubit is given by $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. When a qubit is measured, it collapses to either $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$.

Entanglement

Entanglement is a phenomenon where two or more qubits become correlated in such a way that the state of one qubit cannot be described independently of the state of the others. For example, the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is an entangled state of two qubits. Measuring one qubit instantly determines the outcome of measuring the other, regardless of the distance between them.

Quantum Gates

Quantum gates are the building blocks of quantum circuits. They are represented by unitary matrices and act on one or more qubits. Some common gates include the Hadamard gate, the Pauli-X gate, and the CNOT gate. The Hadamard gate creates superposition, while the CNOT gate can entangle two qubits.

Chapter 2

Quantum algorithms exploit the unique properties of quantum mechanics to solve problems more efficiently than classical algorithms. Shor's algorithm can factor large integers exponentially faster than the best known classical algorithm, which has significant implications for cryptography. Grover's algorithm provides a quadratic speedup for searching unsorted databases.

Shor's Algorithm

Shor's algorithm relies on the quantum Fourier transform to find the period of a function. Once the period is found, the factors of the integer can be determined with high probability. The algorithm has a time complexity of $O((\log N)^3)$, where N is the number to be factored.

Grover's Algorithm

Grover's algorithm searches an unsorted database of N items in $O(\sqrt{N})$ time. It uses amplitude amplification to increase the probability of the desired item. The algorithm is optimal, meaning no quantum algorithm can search faster than $O(\sqrt{N})$.

Chapter 3

Quantum error correction is essential for building reliable quantum computers. Qubits are susceptible to noise and decoherence, which can corrupt the information they store. Error correction codes encode logical qubits into multiple physical qubits, allowing errors to be detected and corrected without measuring the logical qubit directly.

Surface Codes

Surface codes are a promising approach to quantum error correction. They arrange qubits on a two-dimensional lattice and use local measurements to detect errors. The threshold for surface codes is relatively high, meaning that they can tolerate a relatively large error rate in the physical qubits.

Fault-Tolerant Quantum Computation

Fault-tolerant quantum computation aims to perform computations even when some components fail. This requires careful design of quantum circuits and error correction protocols. The goal is to keep the error rate of the logical qubits below a certain threshold, so that arbitrarily long computations can be performed reliably.

⠠ ⠏⠇⠁ ⠉⠊ ⠞⠓⠑ ⠞⠗⠁⠙⠑ ⠞⠗⠁⠝⠎⠏⠕⠗⠞⠁⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠉⠗⠑⠁⠞
⠊⠞⠎ ⠞⠗⠥⠑ ⠃⠑⠝⠑⠋⠊⠞⠎ ⠁⠗⠑ ⠞⠕ ⠃⠑ ⠋⠕⠥⠝⠙ ⠕⠝ ⠞⠓⠑ ⠏⠁⠗⠞⠊⠑⠎
⠞⠕ ⠞⠓⠑⠊⠗ ⠉⠕⠝⠞⠗⠁⠉⠞⠥⠁⠇ ⠁⠝⠙ ⠞⠕ ⠞⠓⠑ ⠏⠥⠃⠇⠊⠉
⠞⠗⠥⠎⠞ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠊⠑⠎ ⠺⠓⠕ ⠁⠗⠑ ⠊⠝⠧⠑⠎⠞⠑⠙ ⠊⠝ ⠞⠓⠑⠊⠗
⠍⠁⠝⠁⠛⠑⠙ ⠁⠝⠙ ⠕⠏⠑⠗⠁⠞⠑⠙ ⠋⠁⠊⠗⠇⠽ ⠁⠝⠙ ⠺⠑⠇⠇ ⠺⠊⠞⠓ ⠞⠓⠑
⠏⠗⠕⠧⠊⠎⠊⠕⠝⠎ ⠕⠋ ⠞⠓⠑ ⠁⠉⠞ ⠁⠝⠙ ⠞⠓⠑ ⠗⠥⠇⠑⠎ ⠁⠙⠕⠏⠞⠑⠙ ⠞⠓⠑⠗⠑⠥⠝⠙⠑⠗
⠞⠓⠑ ⠁⠥⠞⠓⠕⠗⠊⠞⠽ ⠕⠋ ⠞⠓⠑ ⠎⠑⠉⠗⠑⠞⠁⠗⠽ ⠕⠋ ⠎⠞⠁⠞⠑ ⠁⠝⠙ ⠞⠓⠑
⠏⠕⠺⠑⠗⠎ ⠉⠕⠝⠋⠑⠗⠗⠑⠙ ⠥⠏⠕⠝ ⠞⠓⠑⠍ ⠃⠽ ⠞⠓⠑ ⠉⠕⠝⠎⠞⠊⠞⠥⠞⠊⠕⠝
⠕⠋ ⠞⠓⠑ ⠥⠝⠊⠞⠑⠙ ⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠁⠍⠑⠗⠊⠉⠁ ⠞⠕ ⠞⠓⠑ ⠑⠋⠋⠑⠉⠞ ⠞⠓⠁⠞
⠞⠓⠑ ⠁⠉⠞ ⠙⠕⠑⠎ ⠝⠕⠞ ⠊⠝⠋⠗⠊⠝⠛⠑ ⠁⠝⠽ ⠏⠁⠞⠑⠝⠞ ⠕⠗ ⠞⠗⠁⠙⠑⠍⠁⠗⠅
⠗⠊⠛⠓⠞⠎ ⠕⠋ ⠁⠝⠽ ⠏⠑⠗⠎⠕⠝ ⠕⠗ ⠁⠝⠽ ⠕⠞⠓⠑⠗ ⠑⠝⠞⠊⠞⠽ ⠺⠓⠁⠞⠎⠕⠑⠧⠑⠗
⠠ ⠠⠠⠠

⠠
⠠⠠⠠

⠠⠠⠠

⠠ ⠠⠠⠠

[illegible]

[illegible]

[illegible] [illegible] [illegible] [

[illegible]

[illegible]

· · · ·

⠠ ⠞⠓⠑⠗⠑ ⠊⠎ ⠝⠕ ⠙⠕⠥⠃⠞ ⠞⠓⠁⠞ ⠊⠝ ⠞⠓⠑ ⠍⠊⠙⠙⠇⠑ ⠕⠋ ⠁ ⠉⠕⠗⠝⠋⠊⠑⠇⠙ ⠁ ⠞⠗⠁⠉⠞ ⠕⠋ ⠇⠁⠝⠙ ⠞⠓⠁⠞ ⠺⠁⠎ ⠝⠕⠞ ⠙⠑⠎⠊⠛⠝⠁⠞⠑⠙ ⠁⠎ ⠁ ⠗⠕⠁⠙ ⠁⠝⠙ ⠝⠕⠞ ⠁ ⠞⠗⠁⠉⠞ ⠕⠋ ⠇⠁⠝⠙ ⠺⠊⠞⠓ ⠁ ⠙⠑⠋⠊⠝⠑⠙ ⠃⠕⠥⠝⠙⠁⠗⠽ ⠁⠝⠙ ⠁ ⠙⠑⠋⠊⠝⠑⠙ ⠙⠑⠎⠞⠊⠝⠁⠞⠊⠕⠝ ⠠⠺⠓⠑⠝ ⠁ ⠗⠕⠁⠙ ⠊⠎ ⠝⠕⠞ ⠁ ⠗⠕⠁⠙ ⠁⠝⠙ ⠝⠕⠞ ⠁ ⠞⠗⠁⠉⠞ ⠕⠋ ⠇⠁⠝⠙ ⠺⠊⠞⠓

⠠⠁⠝⠙ ⠠⠓⠑⠇⠇⠕ ⠊ ⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇ ⠠⠓⠑⠁⠇⠞⠓ ⠠⠕⠗⠛⠁⠝⠊⠵⠁⠞⠊⠕⠝ ⠁⠝⠙ ⠠⠊⠝
⠞⠓⠑ ⠠⠍⠊⠝⠊⠎⠞⠗⠽ ⠕⠋ ⠠⠓⠑⠁⠇⠞⠓ ⠠⠁⠝⠙ ⠠⠓⠥⠍⠁⠝ ⠠⠎⠑⠗⠧⠊⠉⠑⠎ ⠠⠊⠝ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠼⠁ ⠕⠋
⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠝⠁⠞⠊⠕⠝⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠏⠑⠕⠏⠇⠑ ⠕⠋ ⠠⠞⠓⠑
⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠕⠋ ⠠⠏⠓⠽⠎⠊⠉⠁⠇ ⠠⠞⠓⠑⠗⠁⠏⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑
⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠉⠁⠙⠑⠍⠽ ⠕⠋ ⠠⠠⠎⠉⠊⠑⠝⠉⠑⠠⠠ ⠠⠁⠝⠙ ⠠⠠⠍⠑⠙⠊⠉⠊⠝⠑⠠⠠
⠠⠁⠝⠙ ⠠⠠⠎⠥⠗⠛⠑⠗⠽⠠⠠ ⠠⠁⠗⠑ ⠠⠁⠍⠕⠝⠛ ⠠⠞⠓⠑ ⠠⠍⠕⠎⠞ ⠠⠊⠍⠏⠕⠗⠞⠁⠝⠞
⠠⠏⠗⠕⠋⠑⠎⠎⠊⠕⠝⠎ ⠊⠝ ⠠⠞⠓⠑ ⠠⠍⠑⠙⠊⠉⠁⠇ ⠠⠁⠝⠙ ⠠⠎⠥⠗⠛⠊⠉⠁⠇ ⠠⠋⠊⠑⠇⠙⠎⠲
⠠⠠⠎⠑⠑ ⠠⠁⠇⠎⠕ ⠠⠠

⠠⠠⠠⠠⠠ ⠠⠠⠠ ⠠ ⠠⠠⠠⠠ ⠠ ⠠⠠⠠ ⠠⠠⠠⠠ ⠠ ⠠⠠⠠ ⠠ ⠠ ⠠⠠
⠠⠠ ⠠ ⠠⠠⠠ ⠠⠠⠠ ⠠⠠⠠⠠, ⠠⠠ ⠠⠠ ⠠⠠⠠⠠⠠

⠠⠞⠓⠑ ⠠⠁⠝⠝⠁⠇⠎ ⠕⠋ ⠠⠞⠓⠑ ⠠⠺⠊⠝⠙⠎⠕⠺ ⠠⠏⠁⠎⠞ ⠠⠉⠕⠍⠏⠁⠝⠊⠕⠝
⠙⠁⠞⠁ ⠠⠁⠝⠁⠇⠽⠎⠊⠎ ⠠⠞⠕ ⠠⠁⠙⠧⠁⠝⠉⠑ ⠠⠞⠓⠑ ⠠⠎⠞⠁⠞⠑ ⠠⠕⠋ ⠠⠁⠗⠞ ⠠⠁⠝⠙ ⠠⠠⠠

⠠⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠉⠕⠍⠍⠊⠎⠎⠊⠕⠝ ⠕⠋ ⠠⠋⠇⠕⠗⠊⠙⠁ ⠊⠎ ⠞⠓⠑
⠗⠑⠎⠏⠕⠝⠎⠊⠃⠇⠑ ⠋⠕⠗ ⠞⠓⠑ ⠉⠕⠝⠎⠑⠗⠧⠁⠞⠊⠕⠝ ⠕⠋ ⠝⠁⠞⠥⠗⠁⠇ ⠗⠑⠎⠕⠥⠗⠉⠑⠎ ⠁⠝⠙
⠞⠓⠑ ⠎⠥⠎⠞⠁⠊⠝⠁⠃⠇⠑ ⠥⠎⠑ ⠕⠋ ⠞⠓⠑⠊⠗ ⠗⠑⠎⠕⠥⠗⠉⠑⠎ ⠊⠝ ⠋⠁⠗⠍⠊⠝⠛ ⠁⠝⠙
⠋⠕⠗⠑⠎⠞⠗⠽ ⠊⠝ ⠋⠇⠕⠗⠊⠙⠁ ⠁⠝⠙ ⠞⠓⠗⠕⠥⠛⠓⠕⠥⠞ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠋⠇⠕⠗⠊⠙⠁ ⠁⠝⠙
⠞⠓⠑ ⠊⠍⠏⠁⠉⠞ ⠕⠋ ⠞⠓⠑⠊⠗ ⠥⠎⠑ ⠕⠝ ⠞⠓⠑ ⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞ ⠁⠝⠙ ⠊⠞⠎
⠙⠑⠧⠑⠇⠕⠏⠍⠑⠝⠞ ⠊⠝ ⠞⠓⠑ ⠋⠥⠞⠥⠗⠑ ⠕⠋ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠋⠇⠕⠗⠊⠙⠁ ⠁⠝⠙ ⠊⠞⠎
⠉⠕⠥⠝⠞⠊⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞ ⠁⠝⠙ ⠊⠞⠎ ⠗⠑⠎⠕⠥⠗⠉⠑⠎ ⠁⠝⠙ ⠞⠓⠑
⠏⠕⠏⠥⠇⠁⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠋⠇⠕⠗⠊⠙⠁ ⠁⠝⠙ ⠊⠞⠎ ⠉⠕⠥⠝⠞⠊⠑⠎ ⠁⠝⠙
⠞⠓⠑ ⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞ ⠁⠝⠙ ⠊⠞⠎ ⠗⠑⠎⠕⠥⠗⠉⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠏⠕⠏⠥⠇⠁⠞⠊⠕⠝ ⠕⠋
⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠋⠇⠕⠗⠊⠙⠁ ⠁⠝⠙ ⠊⠞⠎ ⠉⠕⠥⠝⠞⠊⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞
⠁⠝⠙ ⠊⠞⠎ ⠗⠑⠎⠕⠥⠗⠉⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠏⠕⠏⠥⠇⠁⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠋⠇⠕⠗⠊⠙⠁
⠁⠝⠙ ⠊⠞⠎ ⠉⠕⠥⠝⠞⠊⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞ ⠁⠝⠙ ⠊⠞⠎ ⠗⠑⠎⠕⠥⠗⠉⠑⠎ ⠁⠝⠙
⠞⠓⠑

⠠⠠⠠⠠

⠠⠃⠗⠁⠊⠇⠇⠑ ⠞⠕ ⠏⠗⠊⠝⠞ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠁⠞ ⠁⠝ ⠁⠗⠃⠊⠞⠗⠁⠗⠽ ⠙⠑⠛⠗⠑⠑
⠕⠋ ⠉⠕⠍⠏⠇⠑⠞⠑⠝⠑⠎⠎

⠠⠁ ⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞ ⠕⠋ ⠃⠁⠎⠊⠉ ⠅⠝⠕⠺⠇⠑⠙⠛⠑ ⠕⠋⠞⠑⠝ ⠉⠕⠝⠎⠊⠎⠞⠎ ⠕⠋
⠁ ⠋⠑⠺ ⠎⠑⠝⠞⠑⠝⠉⠑⠎⠂ ⠊⠝ ⠺⠓⠊⠉⠓ ⠞⠓⠑ ⠎⠞⠥⠙⠑⠝⠞ ⠋⠁⠍⠊⠇⠊⠁⠗⠊⠵⠑⠎ ⠓⠊⠍⠎⠑⠇⠋
⠺⠊⠞⠓ ⠞⠓⠑ ⠎⠞⠁⠝⠙⠁⠗⠙ ⠎⠽⠍⠃⠕⠇⠎ ⠁⠝⠙ ⠞⠑⠗⠍⠊⠝⠕⠇⠕⠛⠽ ⠁⠎ ⠁ ⠙⠑⠎⠉⠗⠊⠏⠞⠊⠧⠑
⠍⠑⠁⠝⠎ ⠕⠋ ⠙⠑⠎⠉⠗⠊⠃⠊⠝⠛ ⠞⠓⠑ ⠎⠞⠗⠥⠉⠞⠥⠗⠑ ⠕⠋ ⠁ ⠍⠕⠇⠑⠉⠥⠇⠑⠂ ⠞⠓⠑
⠎⠞⠗⠥⠉⠞⠥⠗⠑ ⠕⠋ ⠁ ⠉⠗⠽⠎⠞⠁⠇⠂ ⠁⠝⠙ ⠞⠓⠑ ⠍⠑⠞⠓⠕⠙⠎ ⠥⠎⠑⠙ ⠞⠕ ⠙⠑⠞⠑⠗⠍⠊⠝⠑
⠞⠓⠑ ⠎⠞⠗⠥⠉⠞⠥⠗⠑ ⠕⠋ ⠁ ⠍⠁⠛⠝⠑⠞⠊⠉ ⠋⠊⠑⠇⠙⠂ ⠁⠝⠙ ⠁⠇⠎⠕ ⠞⠓⠑ ⠎⠞⠗⠥⠉⠞⠥⠗⠑
⠕⠋ ⠁ ⠉⠗⠽⠎⠞⠁⠇⠲

⠠ ⠠⠍⠊⠉⠓⠁⠑⠇ ⠠⠠⠙⠁⠧⠊⠙⠎ ⠠ ⠠⠉⠕
⠠⠠⠠⠎⠞⠁⠝⠋⠕⠗⠙ ⠠⠠⠍⠁⠗⠅

⠠⠠⠍⠁⠗⠅ ⠠⠠⠇⠁⠥⠗⠁ ⠠⠠⠃⠁⠗⠃⠁⠗⠁ ⠠⠠⠎⠞⠁⠝⠋⠕⠗⠙

⠠⠠⠃⠇⠁⠝⠅ ⠠⠠⠇⠁⠃ ⠠⠠⠏⠗⠑⠎⠎ ⠠⠠⠎⠞⠁⠝⠋⠕⠗⠙ ⠠⠠⠉⠁⠇⠊⠋⠕⠗⠝⠊⠁
⠠⠠⠠⠎⠞⠁⠝⠋⠕⠗⠙ ⠠⠠⠉⠁⠇⠊⠋⠕⠗⠝⠊⠁ ⠠

⠀⠀⠀

⠠⠠
⠠⠠⠠

⠠⠊⠇⠇⠥⠎⠞⠗⠁⠞⠊⠕⠝ ⠠⠁ ⠎⠑⠇⠋ ⠃⠇⠁⠙⠑ ⠉⠁⠝ ⠓⠑⠇⠏ ⠉⠓⠊⠇⠙⠗⠑⠝ ⠞⠁⠅⠑ ⠎⠞⠑⠏⠎ ⠞⠕⠺⠁⠗⠙⠎
⠠⠊⠇⠇⠥⠎⠞⠗⠁⠞⠊⠕⠝ ⠊⠎ ⠁ ⠎⠑⠇⠋ ⠃⠇⠁⠙⠑ ⠉⠁⠝ ⠓⠑⠇⠏ ⠉⠓⠊⠇⠙⠗⠑⠝ ⠞⠁⠅⠑ ⠎⠞⠑⠏⠎ ⠞⠕⠺⠁⠗⠙⠎ ⠁ ⠍⠁⠞⠥⠗⠑ ⠁⠙⠥⠇⠞ ⠁⠝⠙ ⠃⠑⠉⠕⠍⠑
⠎ ⠎⠞⠗⠕⠝⠛⠑⠗ ⠁⠝⠙ ⠍⠕⠗⠑ ⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞⠂ ⠎⠁⠋⠑⠗ ⠁⠝⠙ ⠍⠕⠗⠑ ⠎⠥⠉⠉⠑⠎⠎⠋⠥⠇⠂ ⠍⠁⠗⠗⠊⠑⠙ ⠁⠝⠙ ⠓⠁⠏⠏⠊⠑⠗ ⠞⠓⠁⠝ ⠁ ⠉⠓⠊⠇⠙
⠎ ⠺⠓⠕ ⠙⠊⠑⠙ ⠁⠞ ⠞⠓⠑ ⠁⠛⠑ ⠕⠋ ⠞⠺⠑⠝⠞⠽ ⠕⠗ ⠞⠓⠊⠗⠞⠽⠂ ⠠⠊⠇⠇⠥⠎⠞⠗⠁⠞⠊⠕⠝ ⠊⠎ ⠁ ⠎⠑⠇⠋ ⠃⠇⠁⠙⠑ ⠉⠁⠝ ⠓⠑⠇⠏ ⠉⠓⠊⠇⠙⠗⠑⠝
⠞⠁⠅⠑ ⠞⠓⠑⠊⠗ ⠋⠊⠗⠎⠞ ⠎⠞⠑⠏⠎ ⠞⠕⠺⠁⠗⠙⠎ ⠁ ⠍⠁⠞⠥⠗⠑ ⠁⠙⠥⠇⠞ ⠁⠝⠙ ⠃⠑⠉⠕⠍⠑ ⠎⠞⠗⠕⠝⠛⠑⠗ ⠁⠝⠙ ⠍⠕⠗⠑ ⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞⠂ ⠎⠁⠋⠑⠗
⠁⠝⠙ ⠍⠕⠗⠑ ⠎⠥⠉⠉⠑⠎⠎⠋⠥⠇⠂ ⠍⠁⠗⠗⠊⠑⠙ ⠁⠝⠙ ⠓⠁⠏⠏⠊⠑⠗ ⠞⠓⠁⠝ ⠁ ⠉⠓⠊⠇⠙ ⠺⠓⠕ ⠙⠊⠑⠙ ⠁⠞ ⠞⠓⠑ ⠁⠛⠑ ⠕⠋ ⠞⠺⠑⠝⠞⠽ ⠕⠗
⠞⠓⠊⠗⠞⠽⠲ ⠠⠞⠓⠑⠽ ⠇⠊⠧⠑ ⠁ ⠇⠕⠝⠛ ⠞⠁⠇⠇ ⠺⠊⠞⠓ ⠃⠗⠕⠁⠙ ⠎⠓⠕⠥⠇⠙⠑⠗⠎⠂ ⠠⠓⠁⠧⠑ ⠁ ⠎⠞⠗⠁⠊⠛⠓⠞ ⠃⠁⠉⠅⠲

⠠⠎⠑⠑ ⠁⠇⠎⠕ ⠠⠓⠁⠗⠗⠽⠲⠠⠏⠕⠞⠞⠻⠎ ⠠⠙⠑⠁⠞⠓⠂ ⠠⠁ ⠠⠓⠁⠗⠗⠽⠲⠠⠏⠕⠞⠞⠻ ⠠⠍⠊⠝⠊⠁⠞⠥⠗⠑ ⠠⠙⠑⠁⠞⠓⠂ ⠠⠁
⠠⠓⠁⠗⠗⠽⠲⠠⠏⠕⠞⠞⠻ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠎⠕⠉⠉⠑⠗⠑⠗⠲⠠⠠⠓⠁⠗⠗⠽⠲⠠⠏⠕⠞⠞⠻⠲⠠⠉⠕⠍⠂ ⠠⠁⠝⠙ ⠠⠓⠁⠗⠗⠽⠲⠠⠏⠕⠞⠞⠻
⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠎⠕⠉⠉⠑⠗⠑⠗⠲⠠⠠⠓⠁⠗⠗⠽⠲⠠⠏⠕⠞⠞⠻⠲⠠⠉⠕⠍ ⠠⠁⠗⠑ ⠠⠁⠇⠎⠕ ⠠⠁⠝ ⠠⠁⠍⠁⠵⠕⠝ ⠠⠃⠑⠎⠞⠎⠑⠇⠇⠑⠗⠎

Alexandra Chen
Department of Computer Science
University of Northfield
alexandra.chen@northfield.edu

Marcus Webb
Department of Computer Science
University of Northfield
marcus.webb@northfield.edu

Abstract

Large language models can be prompted to perform tasks they were never explicitly trained on, but the relationship between prompt wording and model behavior is poorly understood. We investigate whether models internally represent the distinction between a task description and the data the task operates on. Using a set of 12 classification tasks, we train linear probes on the residual stream activations of a 7-billion-parameter model and find that task identity is linearly decodable with over 94% accuracy from activations taken before any data tokens appear. Probe accuracy drops sharply once data tokens enter the context, suggesting that the model separates instruction processing from data processing in its early layers. We then show that this separation can be exploited: steering activations along the probe direction changes the model's behavior on the task without altering the data, and steering along the data direction changes the data interpretation without altering the task. These results provide evidence that instruction-following models develop a structured internal representation of what they are being asked to do, and that this structure can be inspected and intervened on directly.

1 Introduction

Instruction-following models can be given a task in natural language and then applied to many different inputs, but it is unclear how the model keeps track of which part of its input is the instruction and which part is the data. Prior work on in-context learning has largely treated the prompt as a single sequence, without asking whether the model internally distinguishes between the two roles a prompt can play.

We ask a simple question: if we look at the activations of a model partway through a prompt, can we tell what task it has been given, and can we tell whether a given token is part of the task description or part of the data? We find that the answer to both is yes, and that the distinction is concentrated in a small number of layers.

Our contributions are as follows. First, we introduce a probing setup that separates task identity from data content by holding the data fixed across tasks and varying only the instruction. Second, we show that task identity is linearly decodable from activations before the data begins, and that this decodability collapses once data tokens are processed. Third, we demonstrate that the probe direction can be used to steer the model between tasks on the same input, providing a causal test of the representation we found.

2 Related Work

Prior work on probing has established that linear classifiers can recover syntactic and semantic properties from intermediate representations [1, 2]. Our setup differs in that the property we probe for is not a property of the input but of the relationship between the input and the model's own processing.

Work on in-context learning has studied how models use demonstrations to infer a task [3, 4]. We instead study zero-shot instruction following, where no demonstrations are present, and ask what internal state corresponds to the instruction alone.

Activation steering has been used to control model behavior by adding vectors to the residual stream [5]. We use steering as a diagnostic rather than a control method, testing whether the directions found by probing are causally relevant.

3 Method

We use a 7-billion-parameter decoder-only transformer with 32 layers. For each task we write a short instruction and pair it with a fixed set of 200 data examples drawn from a held-out corpus. We record the residual stream activation at the final token of the instruction, before any data tokens are added, and at every layer.

For each layer we train a logistic regression probe to predict task identity from these activations, using 10-fold cross-validation. We also train probes on activations taken after the data has been appended, and on activations at individual data tokens, to measure how much task information remains once the model is processing data.

To test causality, we compute the mean activation difference between pairs of tasks and add a scaled version of this vector to the residual stream during generation. We measure whether the model's outputs change in the direction of the target task while the data remains unchanged.

4 Results

Probe accuracy at the instruction-final token exceeds 94% in layers 8 through 20, with a peak of 96.3% at layer 14. Accuracy at the first data token drops to near chance in the same layers, indicating that the task representation is largely overwritten once data arrives.

Steering at layer 14 with a coefficient of 4.0 changes the model's output on 71% of examples in the direction of the target task, while leaving the data-dependent content of the output largely intact. Steering along a direction orthogonal to the task probe has no measurable effect on task behavior.

5 Conclusion

We find that instruction-following models represent task identity in a linearly decodable form that is distinct from the representation of the data they operate on, and that this separation is concentrated in the early-to-middle layers. The probe direction we recover can be used to steer the model between tasks on the same input, which we take as evidence that the representation is causally relevant rather than merely correlated with behavior.

References

[1] Hewitt, J., and Manning, C. D. A structural probe for finding syntax in word representations. NAACL, 2019.

[2] Tenney, I., Das, D., and Pavlick, E. BERT rediscovers the classical NLP pipeline. ACL, 2019.

[3] Brown, T. B., et al. Language models are few-shot learners. NeurIPS, 2020.

[4] Min, S., et al. Rethinking the role of demonstrations: what makes in-context learning work? EMNLP, 2022.

[5] Turner, A., et al. Activation addition: steering language models without optimization. arXiv, 2023.

⠠⠠⠠

⠠⠠
⠠⠠⠠

⠠⠓⠁⠎ ⠞⠓⠑ ⠍⠕⠎⠞ ⠙⠑⠞⠁⠊⠇⠑⠙ ⠊⠝⠋⠕⠗⠍⠁⠞⠊⠕⠝ ⠁⠃⠕⠥⠞ ⠞⠓⠑
⠊⠗ ⠏⠗⠕⠙⠥⠉⠞⠎ ⠁⠝⠙ ⠞⠓⠑⠊⠗ ⠏⠗⠊⠉⠑⠎ ⠞⠕ ⠞⠓⠑ ⠏⠥⠃⠇⠊⠉
⠁⠝⠙ ⠏⠗⠊⠧⠁⠞⠑ ⠍⠁⠗⠅⠑⠞⠎ ⠁⠗⠑ ⠁⠧⠁⠊⠇⠁⠃⠇⠑ ⠞⠕ ⠁⠇⠇ ⠞⠓⠕⠎⠑
⠺⠓⠕ ⠓⠁⠧⠑ ⠁ ⠙⠑⠎⠊⠗⠑ ⠞⠕ ⠏⠥⠗⠉⠓⠁⠎⠑ ⠁ ⠉⠕⠏⠽ ⠕⠋ ⠞⠓⠊⠎
⠺⠕⠗⠅ ⠺⠊⠞⠓⠕⠥⠞ ⠊⠝⠉⠥⠗⠗⠊⠝⠛ ⠁ ⠏⠁⠽⠍⠑⠝⠞ ⠕⠗ ⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇
⠉⠕⠎⠞⠎ ⠕⠋ ⠁⠝⠽ ⠅⠊⠝⠙ ⠕⠋ ⠁⠙⠧⠑⠗⠞⠊⠎⠊⠝⠛ ⠕⠗ ⠎⠏⠁⠍⠍⠊⠝⠛ ⠊⠎
⠞⠕ ⠙⠊⠎⠉⠕⠝⠞⠊⠝⠥⠑ ⠞⠓⠑ ⠎⠑⠗⠧⠊⠉⠑ ⠁⠞ ⠁⠝⠽ ⠞⠊⠍⠑ ⠁⠝⠙ ⠞⠕
⠞⠁⠅⠑ ⠁⠝⠽ ⠁⠉⠞⠊⠕⠝ ⠞⠓⠁⠞ ⠺⠑ ⠙⠑⠑⠍ ⠝⠑⠉⠑⠎⠎⠁⠗⠽ ⠕⠗ ⠁⠙⠧⠊⠎⠁⠃⠇⠑
⠠⠞⠓⠑ ⠏⠁⠗⠞⠽ ⠓⠁⠎ ⠝⠕

⠠⠠⠞⠓⠑ ⠠⠞⠓⠑⠗⠍⠁⠇ ⠠⠠⠏⠗⠕⠏⠑⠗⠞⠊⠑⠎ ⠠⠠⠞⠓⠑ ⠠⠞⠓⠑⠗⠍⠁⠇ ⠠⠠⠏⠗⠕⠏⠑⠗⠞⠊⠑⠎ ⠠⠠⠞⠓⠑
⠠⠠⠞⠓⠑⠗⠍⠁⠇ ⠠⠠⠏⠗⠕⠏⠑⠗⠞⠊⠑⠎ ⠠⠠⠞⠓⠑ ⠠

⠠⠍ ⠠⠠⠁⠠⠠ ⠠⠠⠠⠁ ⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠ ⠠⠠⠠⠠ ⠠⠠⠠⠠ ⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠
⠠⠠⠠⠠⠠⠠

⠠⠠⠠

⠠⠤⠉ ⠉⠉⠉

[illegible]

[illegible]

[illegible]

⠠⠝⠑⠺ ⠽⠕⠗⠅ ⠉⠊⠞⠽ ⠉⠕⠥⠝⠉⠊⠇ ⠞⠕ ⠁⠙⠙⠗⠑⠎⠎ ⠁ ⠙⠁⠝⠛⠑⠗⠕⠥⠎
⠎⠊⠞⠥⠁⠞⠊⠕⠝ ⠺⠓⠑⠗⠑ ⠁ ⠉⠓⠊⠇⠙ ⠊⠎ ⠃⠑⠊⠝⠛ ⠑⠝⠙⠁⠝⠛⠑⠗⠑⠙ ⠃⠽ ⠁
⠇⠁⠗⠛⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠕⠗ ⠛⠕⠧⠑⠗⠝⠍⠑⠝⠞ ⠁⠛⠑⠝⠉⠽ ⠞⠓⠁⠞ ⠊⠎ ⠞⠁⠅⠊⠝⠛ ⠁⠉⠞⠊⠕⠝
⠞⠕ ⠏⠗⠕⠞⠑⠉⠞ ⠞⠓⠑ ⠉⠓⠊⠇⠙ ⠋⠗⠕⠍ ⠁ ⠏⠕⠞⠑⠝⠞⠊⠁⠇⠇⠽

⠠⠃
⠠⠠⠠

⠠⠞⠓⠑ ⠠⠁⠗⠞ ⠕⠋ ⠠⠙⠑⠍⠕⠉⠗⠁⠉⠽ ⠊⠎ ⠞⠓⠁⠞ ⠞⠓⠑ ⠏⠑⠕⠏⠇⠑ ⠺⠓⠕ ⠁⠗⠑ ⠺⠊⠇⠇⠊⠝⠛ ⠞⠕ ⠙⠊⠑ ⠋⠕⠗ ⠊⠞⠎ ⠊⠙⠑⠁⠇⠎ ⠁⠝⠙ ⠁⠗⠑ ⠝⠕⠞ ⠁⠋⠗⠁⠊⠙ ⠞⠕ ⠇⠊⠧⠑ ⠋⠕⠗ ⠊⠞⠎ ⠊⠙⠑⠁⠇⠎ ⠁⠝⠙

...

...

...

...

...

...

...

...

...

...

⠠⠁⠞⠞⠑⠝⠙⠁⠝⠞ ⠺⠁⠎ ⠙⠊⠎⠏⠁⠞⠉⠓⠑⠙ ⠞⠕ ⠞⠓⠑ ⠉⠁⠎⠑ ⠺⠊⠞⠓ ⠁ ⠉⠕⠍⠏⠇⠁⠊⠝⠞ ⠁⠛⠁⠊⠝⠎⠞ ⠞⠓⠑ ⠏⠕⠇⠊⠉⠑ ⠕⠋⠋⠊⠉⠑⠗ ⠺⠓⠕ ⠁⠗⠗⠊⠧⠑⠙ ⠁⠞ ⠞⠓⠑ ⠎⠉⠑⠝⠑ ⠕⠋ ⠁ ⠙⠕⠍⠑⠎⠞⠊⠉ ⠙⠊⠎⠞⠥⠗⠃⠁⠝⠉⠑ ⠁⠝⠙ ⠞⠁⠅⠑⠝ ⠓⠊⠍ ⠞⠕ ⠞⠓⠑ ⠏⠕⠇⠊⠉⠑ ⠎⠞⠁⠞⠊⠕⠝ ⠋⠕⠗ ⠋⠥⠗⠞⠓⠑⠗ ⠊⠝⠧⠑⠎⠞⠊⠛⠁⠞⠊⠕⠝⠲

⠠⠞⠓⠑ ⠕⠋⠋⠊⠉⠑⠗ ⠺⠁⠎ ⠁⠇⠎⠕ ⠊⠙⠑⠝⠞⠊⠋⠊⠑⠙ ⠁⠎ ⠁ ⠏⠕⠎⠎⠊⠃⠇⠑ ⠺⠊⠞⠝⠑⠎⠎ ⠕⠋ ⠞⠓⠑ ⠁⠉⠉⠊⠙⠑⠝⠞⠲ ⠠⠞⠓⠑ ⠙⠊⠎⠏⠁⠞⠉⠓⠑⠗ ⠁⠎⠅⠑⠙ ⠁ ⠉⠕⠍⠏⠁⠝⠊⠕⠝ ⠺⠓⠕ ⠓⠁⠙ ⠁⠇⠗⠑⠁⠙⠽ ⠗⠑⠎⠏⠕⠝⠙⠑⠙ ⠁⠝⠙ ⠺⠁⠎ ⠁⠇⠎⠕ ⠁⠎⠅⠑⠙ ⠞⠕ ⠎⠞⠁⠽ ⠁⠝⠙ ⠞⠁⠅⠑ ⠞⠓⠑ ⠏⠕⠇⠊⠉⠑ ⠎⠞⠁⠞⠊⠕⠝ ⠋⠕⠗ ⠋⠥⠗⠞⠓⠑⠗ ⠊⠝⠧⠑⠎⠞⠊⠛⠁⠞⠊⠕⠝⠲

⠠⠠⠠

⠠⠞⠓⠑ ⠠⠠⠉⠕⠍⠏⠁⠝⠽ ⠊⠎ ⠁ ⠏⠥⠃⠇⠊⠉⠇⠽ ⠞⠗⠁⠙⠑⠙ ⠉⠕⠍⠏⠁⠝⠽ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠊⠝⠉⠊⠏⠁⠇ ⠕⠉⠉⠥⠏⠁⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠊⠎ ⠁ ⠙⠊⠧⠑⠗⠎⠊⠋⠊⠑⠙ ⠉⠕⠍⠃⠊⠝⠁⠞⠑ ⠕⠋ ⠎⠑⠧⠑⠗⠁⠇ ⠎⠞⠗⠁⠞⠑⠛⠊⠑⠎ ⠁⠝⠙ ⠞⠁⠗⠛⠑⠞ ⠍⠁⠗⠅⠑⠞⠎ ⠁⠝⠙ ⠊⠎ ⠞⠓⠑ ⠇⠁⠗⠛⠑⠎⠞ ⠏⠗⠊⠧⠁⠞⠑⠇⠽ ⠓⠑⠇⠙ ⠉⠕⠍⠏⠁⠝⠽ ⠊⠝ ⠞⠓⠑ ⠺⠕⠗⠇⠙⠲ ⠠⠠⠉⠕⠍⠏⠁⠝⠽ ⠓⠑⠁⠙⠟⠥⠁⠗⠞⠑⠗⠎ ⠁⠗⠑ ⠊⠝ ⠠⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠠⠉⠊⠞⠽⠂ ⠠⠠⠉⠁⠇⠊⠋⠕⠗⠝⠊⠁⠲ ⠠⠠⠉⠕⠍⠏⠁⠝⠽ ⠙⠑⠞⠁⠊⠇⠎ ⠁⠗⠑ ⠁⠧⠁⠊⠇⠁⠃⠇⠑ ⠁⠞ ⠓⠞⠞⠏⠎⠒⠸⠌⠸⠌⠺⠺⠺⠲⠉⠕⠍⠏⠁⠝⠽⠲⠉⠕⠍⠸⠌⠁⠃⠕⠥⠞⠸⠌⠲

…

…

…

⠠⠁⠎⠎⠊⠎⠞ ⠍⠽ ⠁ ⠎⠞⠁⠝⠙⠁⠗⠙ ⠓⠁⠝⠙⠺⠗⠊⠞⠞⠑⠝ ⠁⠉⠉⠕⠥⠝⠞ ⠊⠝ ⠁ ⠉⠕⠍⠏⠁⠝⠽
⠞⠓⠁⠞ ⠊⠎ ⠎⠑⠞ ⠥⠏ ⠞⠕ ⠏⠗⠕⠙⠥⠉⠑ ⠁ ⠎⠞⠁⠞⠑⠍⠑⠝⠞ ⠕⠝ ⠞⠓⠑ ⠎⠞⠁⠞⠥⠎
⠕⠋ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠁⠎ ⠁ ⠺⠓⠕⠇⠑ ⠁⠝⠙ ⠊⠞ ⠊⠎ ⠞⠓⠑ ⠗⠑⠎⠏⠕⠝⠎⠊⠃⠊⠇⠊⠞⠽
⠕⠋ ⠞⠓⠕⠎⠑ ⠺⠓⠕ ⠁⠉⠞ ⠕⠝ ⠊⠞⠎ ⠃⠑⠓⠁⠇⠋ ⠞⠕ ⠞⠓⠑ ⠑⠋⠋⠑⠉⠞ ⠞⠓⠁⠞
⠊⠞ ⠓⠁⠎ ⠕⠝ ⠞⠓⠑ ⠺⠕⠗⠇⠙ ⠁⠝⠙ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠊⠞⠎ ⠎⠓⠁⠗⠑⠓⠕⠇⠙⠑⠗⠎
⠁⠗⠑ ⠁⠇⠎⠕ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠕⠋ ⠞⠓⠑ ⠍⠕⠎⠞ ⠙⠊⠋⠋⠊⠉⠥⠇⠞ ⠞⠽⠏⠑ ⠕⠋ ⠏⠑⠕⠏⠇⠑
⠞⠕ ⠋⠊⠝⠙⠲ ⠠⠁⠎⠎⠊⠎⠞ ⠍⠽ ⠁ ⠎⠞⠁⠝⠙⠁⠗⠙ ⠓⠁⠝⠙⠺⠗⠊⠞⠞⠑⠝ ⠁⠉⠉⠕⠥⠝⠞
⠊⠝ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠁⠎ ⠁ ⠺⠓⠕⠇⠑ ⠁⠝⠙ ⠊⠞ ⠊⠎ ⠞⠓⠑ ⠗⠑⠎⠏⠕⠝⠎⠊⠃⠊⠇⠊⠞⠽
⠕⠋ ⠞⠓⠕⠎⠑ ⠺⠓⠕ ⠁⠉⠞ ⠕⠝ ⠊⠞⠎ ⠃⠑⠓⠁⠇⠋ ⠞⠕ ⠞⠓⠑ ⠑⠋⠋⠑⠉⠞ ⠞⠓⠁⠞
⠊⠞ ⠓⠁⠎ ⠕⠝ ⠞⠓⠑ ⠺⠕⠗⠇⠙ ⠁⠝⠙ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠊⠞⠎ ⠎⠓⠁⠗⠑⠓⠕⠇⠙⠑⠗⠎
⠁⠗⠑ ⠁⠇⠎⠕ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠕⠋ ⠞⠓⠑ ⠍⠕⠎⠞ ⠙⠊⠋⠋⠊⠉⠥⠇⠞ ⠞⠽⠏⠑ ⠕⠋ ⠏⠑⠕⠏⠇⠑
⠞⠕ ⠋⠊⠝⠙⠲ ⠠⠁⠎⠎⠊⠎⠞ ⠍⠽ ⠁ ⠎⠞⠁⠝⠙⠁⠗⠙ ⠓⠁⠝⠙⠺⠗⠊⠞⠞⠑⠝ ⠁⠉⠉⠕⠥⠝⠞
⠊⠝ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠁⠎ ⠁ ⠺⠓⠕⠇⠑ ⠁⠝⠙ ⠊⠞ ⠊⠎ ⠞⠓⠑ ⠗⠑⠎⠏⠕⠝⠎⠊⠃⠊⠇⠊⠞⠽
⠕⠋ ⠞⠓⠕⠎⠑ ⠺⠓⠕ ⠁⠉⠞ ⠕⠝ ⠊⠞⠎ ⠃⠑⠓⠁⠇⠋ ⠞⠕ ⠞⠓⠑ ⠑⠋⠋⠑⠉⠞ ⠞⠓⠁⠞
⠊⠞ ⠓⠁⠎ ⠕⠝ ⠞⠓⠑ ⠺⠕⠗⠇⠙ ⠁⠝⠙ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠊⠞⠎ ⠎⠓⠁⠗⠑⠓⠕⠇⠙⠑⠗⠎
⠁⠗⠑ ⠁⠇⠎⠕ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠕⠋ ⠞⠓⠑ ⠍⠕⠎⠞ ⠙⠊⠋⠋⠊⠉⠥⠇⠞ ⠞⠽⠏⠑ ⠕⠋ ⠏⠑⠕⠏⠇⠑
⠞⠕ ⠋⠊⠝⠙⠲

⠠⠠
⠠⠠⠠

⠀⠀⠀

⠠⠠⠠⠠⠠⠠⠠ ⠠⠠⠠ ⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠
⠠⠠⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠⠠⠠

⠠⠞⠓⠑ ⠠⠙⠁⠗⠅ ⠠⠁⠝⠙ ⠠⠞⠓⠑
⠠⠍⠊⠎⠎⠊⠝⠛ ⠠⠕⠋ ⠠⠎⠁⠍

⠠⠍⠁⠗⠉⠕ ⠠⠙⠑ ⠠⠇⠁ ⠠⠉⠁⠎⠁

⠠⠍⠁⠗⠉⠕ ⠠⠙⠑ ⠠⠇⠁ ⠠⠉⠁⠎⠁ ⠠⠞⠁⠍⠃⠊⠑⠝ ⠠⠞⠁⠍⠃⠊⠑⠝ ⠠⠎⠑ ⠠⠇⠑⠧⠁⠝⠞⠁⠃⠁ ⠠⠞⠁⠍⠃⠊⠑⠝ ⠠⠎⠑ ⠠⠇⠁ ⠠⠉⠁⠎⠁ ⠠⠙⠑

⠠⠁ ⠠⠁⠠⠃⠠⠗⠠⠁⠠⠍ ⠠⠞⠠⠕ ⠠⠃⠠⠁⠠⠇⠠⠁⠠⠝⠠⠉⠠⠑ ⠠⠁ ⠠⠞⠠⠓⠠⠑⠠⠗⠠⠑ ⠠⠁ ⠠⠝⠠⠕
⠠⠠⠁⠠⠁⠠⠁
⠠⠠⠁⠠⠁⠠⠁ ⠠⠠⠁⠠⠁⠠⠁

⠠⠞⠑⠎⠞⠑ ⠙⠑ ⠠⠠⠠⠠⠠⠠

. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

: Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

⠠⠞⠓⠑ ⠠⠋⠊⠗⠎⠞ ⠠⠍⠁⠚⠕⠗ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠞⠕ ⠠⠎⠑⠞ ⠠⠥⠏ ⠠⠁ ⠠⠞⠗⠥⠎⠞
⠠⠋⠥⠝⠙ ⠕⠋ ⠠⠁ ⠠⠙⠊⠎⠞⠊⠝⠉⠞ ⠠⠉⠇⠁⠎⠎ ⠠⠕⠋ ⠠⠎⠑⠗⠧⠊⠉⠑⠎ ⠠⠞⠓⠁⠞
⠠⠺⠕⠥⠇⠙ ⠠⠃⠑ ⠠⠞⠗⠥⠎⠞⠑⠙ ⠠⠞⠕ ⠠⠓⠕⠇⠙ ⠠⠁⠝⠙ ⠠⠍⠁⠝⠁⠛⠑ ⠠⠁
⠠⠏⠕⠗⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠏⠥⠃⠇⠊⠉ ⠠⠁⠉⠉⠕⠥⠝⠞⠎ ⠠⠁⠉⠞⠊⠧⠊⠞⠽
⠠⠞⠓⠁⠞ ⠠⠺⠁⠎ ⠠⠝⠕⠞ ⠠⠁⠥⠞⠓⠕⠗⠊⠵⠑⠙ ⠠⠃⠽ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽
⠠⠁⠝⠙ ⠠⠺⠁⠎ ⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑ ⠠⠁ ⠠⠍⠁⠚⠕⠗ ⠠⠊⠝⠉⠕⠍⠑ ⠠⠁⠝⠙
⠠⠁ ⠠⠍⠁⠚⠕⠗ ⠠⠎⠕⠥⠗⠉⠑ ⠠⠕⠋ ⠠⠞⠁⠭ ⠠⠗⠑⠧⠑⠝⠥⠑ ⠠⠋⠕⠗
⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠋⠕⠗ ⠠⠞⠁⠭ ⠠⠏⠥⠗⠏⠕⠎⠑⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑
⠠⠞⠓⠑ ⠠⠁⠥⠙⠊⠞ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠺⠁⠎ ⠠⠉⠁⠗⠗⠊⠑⠙ ⠠⠕⠥⠞
⠠⠃⠽ ⠠⠞⠓⠑ ⠠⠋⠊⠗⠍ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛ ⠠⠋⠊⠗⠍ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞
⠠⠋⠗⠕⠍ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠁⠞ ⠠⠞⠓⠑ ⠠⠁⠥⠙⠊⠞ ⠠⠺⠁⠎
⠠⠏⠑⠗⠋⠕⠗⠍⠑⠙ ⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠺⠊⠞⠓ ⠠⠁⠥⠙⠊⠞⠊⠝⠛ ⠠⠎⠞⠁⠝⠙⠁⠗⠙⠎
⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠁⠥⠙⠊⠞⠕⠗⠎ ⠠⠁⠗⠑ ⠠⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞ ⠠⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠁⠗⠑ ⠠⠝⠕⠞ ⠠⠊⠝⠧⠕⠇⠧⠑⠙ ⠠⠊⠝ ⠠⠞⠓⠑ ⠠⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞
⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠁⠉⠉⠕⠥⠝⠞⠎ ⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑
⠠⠺⠊⠞⠓ ⠠⠛⠑⠝⠑⠗⠁⠇⠇⠽ ⠠⠁⠉⠉⠑⠏⠞⠑⠙ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛ ⠠⠏⠗⠊⠝⠉⠊⠏⠇⠑⠎
⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠁⠗⠑ ⠠⠝⠕⠞ ⠠⠊⠝⠧⠕⠇⠧⠑⠙ ⠠⠊⠝ ⠠⠞⠓⠑
⠠⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠁⠉⠉⠕⠥⠝⠞⠎
⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠺⠊⠞⠓ ⠠⠛⠑⠝⠑⠗⠁⠇⠇⠽ ⠠⠁⠉⠉⠑⠏⠞⠑⠙ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛
⠠⠏⠗⠊⠝⠉⠊⠏⠇⠑⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠁⠗⠑ ⠠⠝⠕⠞ ⠠⠊⠝⠧⠕⠇⠧⠑⠙
⠠⠊⠝ ⠠⠞⠓⠑ ⠠⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗
⠠⠁⠉⠉⠕⠥⠝⠞⠎ ⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠺⠊⠞⠓ ⠠⠛⠑⠝⠑⠗⠁⠇⠇⠽ ⠠⠁⠉⠉⠑⠏⠞⠑⠙
⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛ ⠠⠏⠗⠊⠝⠉⠊⠏⠇⠑⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠁⠗⠑
⠠⠝⠕⠞ ⠠⠊⠝⠧⠕⠇⠧⠑⠙ ⠠⠊⠝ ⠠⠞⠓⠑ ⠠⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽
⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠁⠉⠉⠕⠥⠝⠞⠎ ⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠺⠊⠞⠓ ⠠⠛⠑⠝⠑⠗⠁⠇⠇⠽
⠠⠁⠉⠉⠑⠏⠞⠑⠙ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛ ⠠⠏⠗⠊⠝⠉⠊⠏⠇⠑⠎

⠠⠞⠓⠑ ⠠⠋⠊⠗⠎⠞ ⠠⠍⠁⠚⠕⠗ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠞⠕ ⠠⠎⠑⠞ ⠠⠥⠏ ⠠⠁ ⠠⠞⠗⠥⠎⠞
⠠⠋⠥⠝⠙ ⠕⠋ ⠠⠁ ⠠⠙⠊⠎⠞⠊⠝⠉⠞ ⠠⠉⠇⠁⠎⠎ ⠠⠕⠋ ⠠⠎⠑⠗⠧⠊⠉⠑⠎ ⠠⠞⠓⠁⠞
⠠⠺⠕⠥⠇⠙ ⠠⠃⠑ ⠠⠞⠗⠥⠎⠞⠑⠙ ⠠⠞⠕ ⠠⠓⠕⠇⠙ ⠠⠁⠝⠙ ⠠⠍⠁⠝⠁⠛⠑ ⠠⠁
⠠⠏⠕⠗⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠏⠥⠃⠇⠊⠉ ⠠⠁⠉⠉⠕⠥⠝⠞⠎ ⠠⠁⠉⠞⠊⠧⠊⠞⠽
⠠⠞⠓⠁⠞ ⠠⠺⠁⠎ ⠠⠝⠕⠞ ⠠⠁⠥⠞⠓⠕⠗⠊⠵⠑⠙ ⠠⠃⠽ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽
⠠⠁⠝⠙ ⠠⠺⠁⠎ ⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑ ⠠⠁ ⠠⠍⠁⠚⠕⠗ ⠠⠊⠝⠉⠕⠍⠑ ⠠⠁⠝⠙
⠠⠁ ⠠⠍⠁⠚⠕⠗ ⠠⠎⠕⠥⠗⠉⠑ ⠠⠕⠋ ⠠⠞⠁⠭ ⠠⠗⠑⠧⠑⠝⠥⠑ ⠠⠋⠕⠗
⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠋⠕⠗ ⠠⠞⠁⠭ ⠠⠏⠥⠗⠏⠕⠎⠑⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑
⠠⠞⠓⠑ ⠠⠁⠥⠙⠊⠞ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠺⠁⠎ ⠠⠉⠁⠗⠗⠊⠑⠙ ⠠⠕⠥⠞
⠠⠃⠽ ⠠⠞⠓⠑ ⠠⠋⠊⠗⠍ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛ ⠠⠋⠊⠗⠍ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞
⠠⠋⠗⠕⠍ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠁⠞ ⠠⠞⠓⠑ ⠠⠁⠥⠙⠊⠞ ⠠⠺⠁⠎
⠠⠏⠑⠗⠋⠕⠗⠍⠑⠙ ⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠺⠊⠞⠓ ⠠⠁⠥⠙⠊⠞⠊⠝⠛ ⠠⠎⠞⠁⠝⠙⠁⠗⠙⠎
⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠁⠥⠙⠊⠞⠕⠗⠎ ⠠⠁⠗⠑ ⠠⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞ ⠠⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠁⠗⠑ ⠠⠝⠕⠞ ⠠⠊⠝⠧⠕⠇⠧⠑⠙ ⠠⠊⠝ ⠠⠞⠓⠑ ⠠⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞
⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠁⠉⠉⠕⠥⠝⠞⠎ ⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑
⠠⠺⠊⠞⠓ ⠠⠛⠑⠝⠑⠗⠁⠇⠇⠽ ⠠⠁⠉⠉⠑⠏⠞⠑⠙ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛ ⠠⠏⠗⠊⠝⠉⠊⠏⠇⠑⠎
⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠁⠗⠑ ⠠⠝⠕⠞ ⠠⠊⠝⠧⠕⠇⠧⠑⠙ ⠠⠊⠝ ⠠⠞⠓⠑
⠠⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠁⠉⠉⠕⠥⠝⠞⠎
⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠺⠊⠞⠓ ⠠⠛⠑⠝⠑⠗⠁⠇⠇⠽ ⠠⠁⠉⠉⠑⠏⠞⠑⠙ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛
⠠⠏⠗⠊⠝⠉⠊⠏⠇⠑⠎

⠠⠞⠓⠑ ⠠⠋⠊⠗⠎⠞ ⠠⠍⠁⠚⠕⠗ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠞⠕ ⠠⠎⠑⠞ ⠠⠥⠏ ⠠⠁ ⠠⠞⠗⠥⠎⠞
⠠⠋⠥⠝⠙ ⠕⠋ ⠠⠁ ⠠⠙⠊⠎⠞⠊⠝⠉⠞ ⠠⠉⠇⠁⠎⠎ ⠠⠕⠋ ⠠⠎⠑⠗⠧⠊⠉⠑⠎ ⠠⠞⠓⠁⠞
⠠⠺⠕⠥⠇⠙ ⠠⠃⠑ ⠠⠞⠗⠥⠎⠞⠑⠙ ⠠⠞⠕ ⠠⠓⠕⠇⠙ ⠠⠁⠝⠙ ⠠⠍⠁⠝⠁⠛⠑ ⠠⠁
⠠⠏⠕⠗⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠏⠥⠃⠇⠊⠉ ⠠⠁⠉⠉⠕⠥⠝⠞⠎ ⠠⠁⠉⠞⠊⠧⠊⠞⠽
⠠⠞⠓⠁⠞ ⠠⠺⠁⠎ ⠠⠝⠕⠞ ⠠⠁⠥⠞⠓⠕⠗⠊⠵⠑⠙ ⠠⠃⠽ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽
⠠⠁⠝⠙ ⠠⠺⠁⠎ ⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑ ⠠⠁ ⠠⠍⠁⠚⠕⠗ ⠠⠊⠝⠉⠕⠍⠑ ⠠⠁⠝⠙
⠠⠁ ⠠⠍⠁⠚⠕⠗ ⠠⠎⠕⠥⠗⠉⠑ ⠠⠕⠋ ⠠⠞⠁⠭ ⠠⠗⠑⠧⠑⠝⠥⠑ ⠠⠋⠕⠗
⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠋⠕⠗ ⠠⠞⠁⠭ ⠠⠏⠥⠗⠏⠕⠎⠑⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑
⠠⠞⠓⠑ ⠠⠁⠥⠙⠊⠞ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠺⠁⠎ ⠠⠉⠁⠗⠗⠊⠑⠙ ⠠⠕⠥⠞
⠠⠃⠽ ⠠⠞⠓⠑ ⠠⠋⠊⠗⠍ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛ ⠠⠋⠊⠗⠍ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞
⠠⠋⠗⠕⠍ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠁⠞ ⠠⠞⠓⠑ ⠠⠁⠥⠙⠊⠞ ⠠⠺⠁⠎
⠠⠏⠑⠗⠋⠕⠗⠍⠑⠙ ⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠺⠊⠞⠓ ⠠⠁⠥⠙⠊⠞⠊⠝⠛ ⠠⠎⠞⠁⠝⠙⠁⠗⠙⠎
⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠁⠥⠙⠊⠞⠕⠗⠎ ⠠⠁⠗⠑ ⠠⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞ ⠠⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠁⠗⠑ ⠠⠝⠕⠞ ⠠⠊⠝⠧⠕⠇⠧⠑⠙ ⠠⠊⠝ ⠠⠞⠓⠑ ⠠⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞
⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠁⠉⠉⠕⠥⠝⠞⠎ ⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑
⠠⠺⠊⠞⠓ ⠠⠛⠑⠝⠑⠗⠁⠇⠇⠽ ⠠⠁⠉⠉⠑⠏⠞⠑⠙ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛ ⠠⠏⠗⠊⠝⠉⠊⠏⠇⠑⠎
⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠁⠗⠑ ⠠⠝⠕⠞ ⠠⠊⠝⠧⠕⠇⠧⠑⠙ ⠠⠊⠝ ⠠⠞⠓⠑
⠠⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠁⠉⠉⠕⠥⠝⠞⠎
⠠⠊⠝ ⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠺⠊⠞⠓ ⠠⠛⠑⠝⠑⠗⠁⠇⠇⠽ ⠠⠁⠉⠉⠑⠏⠞⠑⠙ ⠠⠁⠉⠉⠕⠥⠝⠞⠊⠝⠛
⠠⠏⠗⠊⠝⠉⠊⠏⠇⠑⠎

⠀⠀⠀

[illegible]

[illegible]

[illegible]

[

⠠⠞⠓⠑ ⠠⠞⠗⠊⠃⠥⠝⠁⠇ ⠓⠁⠎ ⠝⠕ ⠏⠕⠺⠑⠗ ⠞⠕ ⠊⠍⠏⠕⠎⠑ ⠁ ⠎⠑⠞⠞⠇⠑⠍⠑⠝⠞ ⠕⠝ ⠁
⠏⠁⠗⠞⠽ ⠺⠓⠕ ⠙⠕⠑⠎ ⠝⠕⠞ ⠁⠉⠉⠑⠏⠞ ⠊⠞⠎ ⠙⠑⠉⠊⠎⠊⠕⠝ ⠺⠊⠞⠓⠊⠝
⠞⠓⠗⠑⠑ ⠍⠕⠝⠞⠓⠎ ⠕⠋ ⠎⠥⠉⠓ ⠁ ⠙⠑⠉⠊⠎⠊⠕⠝ ⠍⠁⠽ ⠃⠑ ⠁⠏⠏⠑⠁⠇⠑⠙
⠁⠛⠁⠊⠝⠎⠞ ⠞⠓⠑ ⠞⠗⠊⠃⠥⠝⠁⠇ ⠁⠝⠙ ⠞⠓⠑ ⠙⠑⠉⠊⠎⠊⠕⠝ ⠊⠎ ⠋⠊⠝⠁⠇
⠁⠝⠙ ⠃⠊⠝⠙⠊⠝⠛ ⠕⠝ ⠁⠇⠇ ⠏⠁⠗⠞⠊⠑⠎ ⠞⠕ ⠞⠓⠑ ⠏⠗⠕⠉⠑⠑⠙⠊⠝⠛⠎ ⠕⠋
⠞⠓⠊⠎ ⠉⠁⠎⠑ ⠁⠝⠙ ⠞⠓⠑ ⠗⠑⠍⠁⠊⠝⠙⠑⠗ ⠕⠋ ⠞⠓⠑ ⠗⠑⠉⠕⠗⠙ ⠕⠋ ⠞⠓⠊⠎
⠏⠗⠕⠉⠑⠑⠙⠊⠝⠛ ⠎⠓⠁⠇⠇ ⠃⠑ ⠎⠥⠃⠍⠊⠞⠞⠑⠙ ⠞⠕ ⠞⠓⠑ ⠉⠕⠥⠗⠞ ⠁⠞ ⠁ ⠏⠇⠁⠉⠑
⠁⠝⠙ ⠞⠊⠍⠑ ⠞⠓⠁⠞ ⠎⠓⠁⠇⠇ ⠃⠑ ⠙⠑⠞⠑⠗⠍⠊⠝⠑⠙ ⠃⠽ ⠞⠓⠑ ⠉⠕⠥⠗⠞⠂

⠠ ⠠⠞⠓⠑⠕⠗⠽ ⠕⠋ ⠠⠉⠕⠍⠏⠁⠗⠁⠞⠊⠧⠑ ⠠⠉⠕⠎⠞⠎
⠠⠠ ⠠⠞⠓⠑⠕⠗⠽ ⠕⠋ ⠠⠉⠕⠍⠏⠁⠗⠁⠞⠊⠧⠑ ⠠⠉⠕⠎⠞⠎

⠠⠠⠠

⠠⠞⠓⠑ ⠙⠁⠝⠉⠑ ⠞⠓⠑⠍⠑ ⠺⠁⠎ ⠁ ⠉⠕⠍⠃⠊⠝⠁⠞⠊⠕⠝ ⠕⠋ ⠉⠕⠥⠝⠞⠗⠽ ⠙⠁⠝⠉⠑ ⠁⠝⠙ ⠎⠕⠍⠑ ⠕⠋ ⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠕⠏⠥⠇⠁⠗ ⠙⠁⠝⠉⠑ ⠞⠓⠑⠍⠑⠎ ⠕⠋ ⠞⠓⠑ ⠞⠊⠍⠑ ⠺⠑⠗⠑ ⠞⠓⠕⠎⠑ ⠞⠓⠁⠞ ⠉⠁⠍⠑ ⠋⠗⠕⠍ ⠞⠓⠑ ⠎⠕⠥⠞⠓⠺⠑⠎⠞ ⠏⠁⠗⠞ ⠕⠋ ⠞⠓⠑ ⠉⠕⠥⠝⠞⠗⠽ ⠁⠝⠙

⠠⠠
⠠⠠⠠

⠠⠁⠗⠞⠊⠉⠇⠑⠎ ⠉⠕⠝⠞⠁⠊⠝ ⠁ ⠎⠞⠗⠥⠉⠞⠥⠗⠑⠙ ⠙⠁⠞⠁⠃⠁⠎⠑ ⠕⠋ ⠞⠓⠑
⠊⠝⠋⠕⠗⠍⠁⠞⠊⠕⠝ ⠞⠓⠁⠞ ⠉⠁⠝ ⠃⠑ ⠗⠑⠁⠙⠊⠇⠽ ⠁⠉⠉⠑⠎⠎⠑⠙ ⠁⠝⠙ ⠏⠗⠕⠉⠑⠎⠎⠑⠙
⠃⠽ ⠁ ⠉⠕⠍⠏⠥⠞⠑⠗ ⠁⠝⠙ ⠞⠓⠑⠝ ⠁ ⠏⠗⠕⠛⠗⠁⠍ ⠞⠓⠁⠞ ⠉⠁⠝ ⠁⠉⠞
⠥⠏⠕⠝ ⠞⠓⠑ ⠙⠁⠞⠁ ⠞⠕ ⠏⠗⠕⠙⠥⠉⠑ ⠁ ⠙⠑⠎⠊⠗⠑⠙ ⠗⠑⠎⠥⠇⠞⠲ ⠠⠁ ⠉⠕⠍⠏⠥⠞⠑⠗
⠏⠗⠕⠛⠗⠁⠍ ⠊⠎ ⠁ ⠎⠑⠞ ⠕⠋ ⠊⠝⠎⠞⠗⠥⠉⠞⠊⠕⠝⠎ ⠞⠓⠁⠞ ⠞⠑⠇⠇ ⠁ ⠉⠕⠍⠏⠥⠞⠑⠗
⠓⠕⠺ ⠞⠕ ⠉⠁⠗⠗⠽ ⠕⠥⠞ ⠁ ⠙⠑⠎⠊⠗⠑⠙ ⠞⠁⠎⠅⠲ ⠠⠁ ⠏⠗⠕⠛⠗⠁⠍ ⠊⠎ ⠁⠇⠎⠕
⠁ ⠎⠑⠞ ⠕⠋ ⠊⠝⠎⠞⠗⠥⠉⠞⠊⠕⠝⠎ ⠞⠓⠁⠞ ⠞⠑⠇⠇ ⠁ ⠉⠕⠍⠏⠥⠞⠑⠗ ⠓⠕⠺ ⠞⠕ ⠉⠁⠗⠗⠽
⠕⠥⠞ ⠁ ⠙⠑⠎⠊⠗⠑⠙ ⠞⠁⠎⠅⠲ ⠠⠁ ⠏⠗⠕⠛⠗⠁⠍ ⠞⠓⠁⠞ ⠊⠍⠏⠇⠑⠍⠑⠝⠞⠎ ⠁⠇⠇ ⠞⠓⠗⠑⠑
⠉⠕⠍⠏⠕⠝⠑⠝⠞⠎ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠏⠥⠞⠁⠞⠊⠕⠝⠁⠇ ⠍⠕⠙⠑⠇ ⠊⠝ ⠁ ⠎⠊⠝⠛⠇⠑
⠏⠗⠕⠛⠗⠁⠍ ⠊⠎ ⠉⠁⠇⠇⠑⠙ ⠁ ⠞⠁⠎⠅⠲

⠠⠠⠠

⠠⠠
⠠⠠⠠

⠠ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠓⠁⠙ ⠎⠑⠧⠑⠗⠁⠇ ⠉⠕⠍⠏⠁⠝⠊⠑⠎ ⠁⠇⠇⠑⠛⠊⠝⠛ ⠞⠓⠁⠞ ⠞⠓⠑
⠽ ⠞⠓⠑⠽ ⠺⠑⠗⠑ ⠊⠝⠞⠑⠗⠋⠑⠗⠊⠝⠛ ⠺⠊⠞⠓ ⠞⠓⠑ ⠁⠉⠞⠊⠧⠊⠞⠽ ⠕⠋ ⠞⠓⠑
⠊⠗ ⠎⠞⠁⠋⠋ ⠞⠓⠑⠽ ⠉⠕⠝⠎⠊⠙⠑⠗⠑⠙ ⠠⠃⠕⠝⠙⠎ ⠁⠝⠙ ⠠⠎⠓⠁⠗⠑⠎ ⠁⠎ ⠁
⠏⠁⠗⠞ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠞⠕ ⠊⠝⠉⠗⠑⠁⠎⠑ ⠞⠓⠑⠊⠗ ⠧⠁⠇⠥⠑⠎ ⠁⠝⠙
⠞⠓⠑⠗⠑⠃⠽ ⠊⠝⠉⠗⠑⠁⠎⠑ ⠞⠓⠑⠊⠗ ⠍⠁⠗⠅⠑⠞ ⠉⠁⠏⠊⠞⠁⠇⠊⠵⠁⠞⠊⠕⠝⠲
⠠⠠⠠

⠠⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠙⠕⠑⠎ ⠝⠕⠞ ⠙⠊⠗⠑⠉⠞⠇⠽ ⠏⠗⠕⠙⠥⠉⠑ ⠁⠝⠽
⠞⠓⠊⠝⠛ ⠞⠓⠁⠞ ⠺⠑ ⠎⠑⠇⠇ ⠕⠗ ⠎⠑⠗⠧⠊⠉⠑⠎⠲ ⠠⠺⠑ ⠙⠕ ⠁ ⠎⠞⠗⠕⠝⠛
⠍⠁⠝ ⠙⠕ ⠞⠓⠁⠞⠲ ⠠⠺⠑ ⠛⠥⠊⠙⠑ ⠁ ⠛⠗⠕⠥⠏ ⠕⠋ ⠛⠗⠑⠁⠞ ⠏⠑⠕⠏⠇⠑⠲ ⠠⠞⠓⠑⠽
⠞⠁⠅⠑ ⠞⠓⠑⠊⠗ ⠁⠙⠧⠊⠉⠑ ⠁⠝⠙ ⠏⠗⠑⠏⠁⠗⠑ ⠞⠓⠑⠍ ⠞⠕ ⠃⠑⠉⠕⠍⠑ ⠞⠓⠑⠊⠗
⠏⠗⠕⠋⠑⠎⠎⠊⠕⠝⠁⠇ ⠉⠕⠝⠎⠥⠇⠞⠁⠝⠞⠎⠲⠠⠠⠠

THEOREM 1. LET f BE A FUNCTION OF ONE VARIABLE WHICH IS DEFINED FOR ALL REAL VALUES OF x AND SUCH THAT $f(x) > x$ FOR ALL x, AND LET f^n DENOTE THE nTH ITERATE OF f. THEN THE FUNCTION F DEFINED BY $F(x) = f^{x+1}(x)$ IS STRICTLY INCREASING.

PROOF. FOR ANY x WE HAVE $F(x+1) = f^{x+2}(x+1) = f^{x+1}(f(x+1)) > f^{x+1}(x+1) > f^{x+1}(x) = F(x)$, SINCE f^{x+1} IS ITSELF STRICTLY INCREASING. THUS F IS STRICTLY INCREASING. $\square$

REMARK. THE HYPOTHESIS $f(x) > x$ CANNOT BE WEAKENED TO $f(x) \geq x$, AS THE IDENTITY FUNCTION SHOWS. NOR CAN IT BE REPLACED BY $f(x) \neq x$, AS $f(x) = x - 1$ SHOWS.

LEMMA 2. IF f IS CONTINUOUS AND $f(x) > x$ FOR ALL x, THEN $f^{n+1}(x) > f^n(x)$ FOR ALL $n \geq 0$ AND ALL x.

PROOF. BY INDUCTION ON n. THE BASE CASE $n = 0$ IS $f(x) > x$. ASSUME $f^n(x) > f^{n-1}(x)$ FOR ALL x. APPLYING f TO BOTH SIDES AND USING THAT f IS INCREASING (A CONSEQUENCE OF CONTINUITY AND $f(x) > x$) GIVES $f^{n+1}(x) > f^n(x)$. $\square$

COROLLARY 3. THE SEQUENCE $x, f(x), f^2(x), \ldots$ IS STRICTLY INCREASING.

EXAMPLE. TAKE $f(x) = x + 1$. THEN $f^n(x) = x + n$, AND $F(x) = f^{x+1}(x) = x + x + 1 = 2x + 1$, WHICH IS STRICTLY INCREASING AS THE THEOREM PREDICTS.

EXAMPLE. TAKE $f(x) = 2x$. THEN $f^n(x) = 2^n x$, AND $F(x) = 2^{x+1} x$. FOR $x = 0$ WE GET $F(0) = 0$, AND FOR $x = 1$ WE GET $F(1) = 4$, SO F IS INCREASING ON THIS PAIR.

EXAMPLE. TAKE $f(x) = x + \sin^2(x) + 1$. SINCE $\sin^2(x) \geq 0$ WE HAVE $f(x) > x$, AND f IS CONTINUOUS, SO THE THEOREM APPLIES. THE ITERATES $f^n(x)$ HAVE NO CLOSED FORM, BUT $F(x) = f^{x+1}(x)$ IS STILL STRICTLY INCREASING.

EXAMPLE. TAKE $f(x) = e^x$. THEN $f(x) > x$ FOR ALL x, AND $f^n(x)$ IS THE nTH ITERATE OF THE EXPONENTIAL. THE FUNCTION $F(x) = f^{x+1}(x)$ GROWS EXTREMELY RAPIDLY; FOR $x = 1$ WE HAVE $F(1) = f^2(1) = e^e \approx 15.154$, AND FOR $x = 2$ WE HAVE $F(2) = f^3(2) = e^{e^e} \approx 3.814 \times 10^6$.

EXAMPLE. TAKE $f(x) = x + 1/x$ FOR $x > 0$. THEN $f(x) > x$, AND f IS CONTINUOUS ON $(0, \infty)$. THE ITERATES $f^n(x)$ GROW LIKE $\sqrt{2n}$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $\sqrt{2x^2} = x\sqrt{2}$.

EXAMPLE. TAKE $f(x) = x + \log(1 + x)$ FOR $x > -1$. THEN $f(x) > x$ FOR ALL $x > -1$, AND f IS CONTINUOUS ON $(-1, \infty)$. THE ITERATES $f^n(x)$ GROW LIKE $\log n$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $x \log x$.

EXAMPLE. TAKE $f(x) = x + 1/(1 + x^2)$. THEN $f(x) > x$ FOR ALL x, AND f IS CONTINUOUS. THE ITERATES $f^n(x)$ GROW LIKE $\sqrt{n}$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $\sqrt{x^2} = x$.

EXAMPLE. TAKE $f(x) = x + \arctan(x) + \pi/2$. THEN $f(x) > x$ FOR ALL x, AND f IS CONTINUOUS. THE ITERATES $f^n(x)$ GROW LIKE $n\pi/2$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $x\pi/2$.

EXAMPLE. TAKE $f(x) = x + 1/\log(x + 2)$ FOR $x > -1$. THEN $f(x) > x$ FOR ALL $x > -1$, AND f IS CONTINUOUS ON $(-1, \infty)$. THE ITERATES $f^n(x)$ GROW LIKE $\sqrt{2n \log n}$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $x\sqrt{2 \log x}$.

EXAMPLE. TAKE $f(x) = x + \log(1 + e^x)$. THEN $f(x) > x$ FOR ALL x, AND f IS CONTINUOUS. THE ITERATES $f^n(x)$ GROW LIKE $n \log 2$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $x \log 2$.

EXAMPLE. TAKE $f(x) = x + 1/(1 + e^x)$. THEN $f(x) > x$ FOR ALL x, AND f IS CONTINUOUS. THE ITERATES $f^n(x)$ GROW LIKE $\log n$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $x \log x$.

EXAMPLE. TAKE $f(x) = x + \log(1 + \log(1 + x))$ FOR $x > -1$. THEN $f(x) > x$ FOR ALL $x > -1$, AND f IS CONTINUOUS ON $(-1, \infty)$. THE ITERATES $f^n(x)$ GROW LIKE $\log \log n$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $x \log \log x$.

EXAMPLE. TAKE $f(x) = x + 1/(1 + \log(1 + x))$ FOR $x > -1$. THEN $f(x) > x$ FOR ALL $x > -1$, AND f IS CONTINUOUS ON $(-1, \infty)$. THE ITERATES $f^n(x)$ GROW LIKE $\sqrt{2n \log \log n}$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $x\sqrt{2 \log \log x}$.

EXAMPLE. TAKE $f(x) = x + \log(1 + \log(1 + \log(1 + x)))$ FOR $x > -1$. THEN $f(x) > x$ FOR ALL $x > -1$, AND f IS CONTINUOUS ON $(-1, \infty)$. THE ITERATES $f^n(x)$ GROW LIKE $\log \log \log n$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $x \log \log \log x$.

EXAMPLE. TAKE $f(x) = x + 1/(1 + \log(1 + \log(1 + x)))$ FOR $x > -1$. THEN $f(x) > x$ FOR ALL $x > -1$, AND f IS CONTINUOUS ON $(-1, \infty)$. THE ITERATES $f^n(x)$ GROW LIKE $\sqrt{2n \log \log \log n}$ FOR LARGE n, SO $F(x) = f^{x+1}(x)$ GROWS LIKE $x\sqrt{2 \log \log \log x}$.

⠠⠃⠽ ⠞⠓⠑ ⠞⠊⠍⠑ ⠏⠗⠑⠎⠎⠥⠗⠑ ⠊⠎ ⠙⠥⠑ ⠞⠓⠑ ⠞⠊⠍⠑ ⠞⠓⠁⠞ ⠏⠗⠑⠎⠎⠥⠗⠑ ⠊⠎
⠁ ⠞⠊⠍⠑ ⠞⠓⠁⠞ ⠊⠎ ⠙⠥⠑ ⠞⠕ ⠁ ⠋⠕⠗⠉⠑ ⠞⠓⠁⠞ ⠊⠎ ⠁⠏⠏⠇⠊⠑⠙ ⠕⠝ ⠁ ⠎⠥⠗⠋⠁⠉⠑ ⠠⠊⠞ ⠊⠎
⠞⠓⠑ ⠏⠓⠽⠎⠊⠉⠁⠇ ⠋⠕⠗⠉⠑ ⠞⠓⠁⠞ ⠁ ⠃⠕⠙⠽ ⠍⠥⠎⠞ ⠑⠭⠻⠞ ⠕⠝ ⠁ ⠎⠥⠗⠋⠁⠉⠑ ⠠⠊⠞
⠊⠎ ⠞⠓⠑ ⠍⠁⠛⠝⠊⠞⠥⠙⠑ ⠕⠋ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠁⠝⠙ ⠞⠓⠑ ⠙⠊⠎⠞⠁⠝⠉⠑ ⠞⠓⠁⠞ ⠞⠓⠑
⠋⠕⠗⠉⠑ ⠁⠉⠞⠎ ⠕⠝ ⠞⠓⠑ ⠕⠃⠚⠑⠉⠞ ⠁⠇⠕⠝⠛ ⠞⠓⠑ ⠙⠊⠗⠑⠉⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑
⠋⠕⠗⠉⠑ ⠊⠞ ⠁⠏⠏⠇⠊⠑⠎ ⠞⠕ ⠞⠓⠑ ⠕⠃⠚⠑⠉⠞ ⠊⠎ ⠞⠓⠑ ⠍⠁⠛⠝⠊⠞⠥⠙⠑ ⠕⠋ ⠞⠓⠑
⠋⠕⠗⠉⠑ ⠞⠓⠁⠞ ⠁⠉⠞⠎ ⠕⠝ ⠞⠓⠑ ⠕⠃⠚⠑⠉⠞ ⠠⠊⠞ ⠊⠎ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠞⠓⠁⠞ ⠁⠉⠞⠎
⠕⠝ ⠞⠓⠑ ⠕⠃⠚⠑⠉⠞ ⠁⠇⠕⠝⠛ ⠞⠓⠑ ⠙⠊⠗⠑⠉⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠁⠝⠙ ⠞⠓⠑
⠙⠊⠎⠞⠁⠝⠉⠑ ⠞⠓⠁⠞ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠁⠉⠞⠎ ⠕⠝ ⠞⠓⠑ ⠕⠃⠚⠑⠉⠞ ⠊⠎ ⠞⠓⠑
⠍⠁⠛⠝⠊⠞⠥⠙⠑ ⠕⠋ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠁⠞ ⠞⠓⠁⠞ ⠙⠊⠎⠞⠁⠝⠉⠑ ⠠⠊⠞ ⠊⠎ ⠞⠓⠑ ⠍⠁⠛⠝⠊⠞⠥⠙⠑
⠕⠋ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠞⠓⠁⠞ ⠁⠉⠞⠎ ⠕⠝ ⠞⠓⠑ ⠕⠃⠚⠑⠉⠞ ⠁⠇⠕⠝⠛ ⠞⠓⠑ ⠙⠊⠗⠑⠉⠞⠊⠕⠝
⠕⠋ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠁⠝⠙ ⠞⠓⠑ ⠙⠊⠎⠞⠁⠝⠉⠑ ⠞⠓⠁⠞ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠁⠉⠞⠎ ⠕⠝ ⠞⠓⠑
⠕⠃⠚⠑⠉⠞ ⠊⠎ ⠞⠓⠑ ⠍⠁⠛⠝⠊⠞⠥⠙⠑ ⠕⠋ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠁⠞ ⠞⠓⠁⠞ ⠙⠊⠎⠞⠁⠝⠉⠑ ⠠⠊⠞
⠊⠎ ⠞⠓⠑ ⠍⠁⠛⠝⠊⠞⠥⠙⠑ ⠕⠋ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠞⠓⠁⠞ ⠁⠉⠞⠎ ⠕⠝ ⠞⠓⠑ ⠕⠃⠚⠑⠉⠞
⠁⠇⠕⠝⠛ ⠞⠓⠑ ⠙⠊⠗⠑⠉⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠁⠝⠙ ⠞⠓⠑ ⠙⠊⠎⠞⠁⠝⠉⠑ ⠞⠓⠁⠞
⠞⠓⠑ ⠋⠕⠗⠉⠑ ⠁⠉⠞⠎ ⠕⠝ ⠞⠓⠑ ⠕⠃⠚⠑⠉⠞

⠠⠠
⠠⠠⠠

⠠⠓⠑ ⠠⠓⠑⠁⠙ ⠠⠋⠁⠉⠞⠕⠗⠽ ⠠⠃⠕⠎⠞⠕⠝ ⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠷⠠⠠⠁⠠⠠⠾ ⠷⠠⠠⠃⠠⠠⠾ ⠠⠁⠝⠙ ⠷⠠⠠⠉⠠⠠⠾ ⠠⠁⠗⠑ ⠠⠁⠍⠕⠝⠛ ⠠⠞⠓⠑ ⠠⠍⠁⠝⠽ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠞⠓⠁⠞ ⠠⠓⠁⠧⠑ ⠠⠃⠑⠑⠝ ⠠⠉⠗⠑⠁⠞⠑⠙ ⠠⠃⠽ ⠠⠞⠓⠑ ⠠⠋⠁⠉⠞⠕⠗⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠙ ⠠⠏⠗⠕⠙⠥⠉⠞⠎ ⠠⠁⠗⠑ ⠠⠙⠑⠎⠊⠛⠝⠑⠙ ⠠⠁⠝⠙ ⠠⠙⠑⠧⠑⠇⠕⠏⠑⠙ ⠠⠞⠕ ⠠⠍⠑⠑⠞ ⠠⠞⠓⠑ ⠠⠝⠑⠑⠙⠎ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠥⠎⠞⠕⠍⠑⠗⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠍⠁⠗⠅⠑⠞ ⠠⠁⠞ ⠠⠁ ⠠⠏⠗⠊⠉⠑ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠁⠞ ⠠⠁ ⠠⠏⠗⠕⠋⠊⠞ ⠠⠁⠝⠙ ⠠⠁ ⠠⠇⠕⠎⠎ ⠠⠁⠝⠙ ⠠⠁⠇⠇ ⠠⠞⠓⠑ ⠠⠏⠁⠗⠞⠊⠑⠎ ⠠⠊⠝⠧⠕⠇⠧⠑⠙ ⠠⠁⠗⠑ ⠠⠁⠉⠞⠥⠁⠇⠇⠽ ⠠⠍⠁⠅⠊⠝⠛ ⠠⠁ ⠠⠏⠗⠕⠋⠊⠞ ⠠⠁⠝⠙ ⠠⠁ ⠠⠇⠕⠎⠎

⠠⠠⠠

⠠⠠
⠠⠠⠠

⠠⠁ ⠠⠊⠊⠊⠓ ⠠⠠⠍⠠⠍⠠⠊⠠⠉⠠⠃⠠⠁ ⠠⠊⠓⠇⠃⠠ ⠠⠃⠠⠁ ⠠⠃⠠⠊⠠⠍⠠⠃⠠⠁⠠⠉⠠⠁ ⠠
⠇⠠⠃⠠⠃⠠⠁ ⠠⠁ ⠠⠍⠠⠃⠠⠁ ⠠⠃⠠⠁⠠⠍⠠⠃⠠⠁⠠⠉⠠⠃⠠⠁ ⠠⠍⠠⠃⠠⠁ ⠠⠓⠠⠍⠠⠃⠠⠍⠠⠃⠠⠁ ⠠⠠
⠠⠊⠠⠊ ⠠⠁⠠⠇⠠⠃

⠠⠠⠠

⠠⠁ ⠠⠃⠗⠊⠞⠊⠎⠓ ⠠⠉⠕⠇⠥⠍⠃⠊⠁ ⠠⠍⠁⠗⠅⠑⠞⠎ ⠠⠁⠝⠙ ⠠⠃⠗⠊⠞⠊⠎⠓
⠠⠁⠎⠊⠁ ⠠⠃⠥⠗⠑⠁⠥ ⠠⠕⠋ ⠠⠑⠉⠕⠝⠕⠍⠊⠉ ⠠⠁⠝⠁⠇⠽⠎⠊⠎
⠠⠁ ⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇ ⠠⠉⠕⠍⠏⠁⠗⠊⠎⠕⠝ ⠠⠕⠋ ⠠⠍⠁⠗⠅⠑⠞
⠠⠁⠉⠉⠑⠎⠎ ⠠⠁⠝⠙ ⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇ ⠠⠉⠕⠍⠏⠁⠗⠊⠎⠕⠝
⠠⠁⠎⠊⠁ ⠠⠃⠥⠗⠑⠁⠥ ⠠⠕⠋ ⠠⠑⠉⠕⠝⠕⠍⠊⠉ ⠠⠁⠝⠁⠇⠽⠎⠊⠎
⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇ ⠠⠉⠕⠍⠏⠁⠗⠊⠎⠕⠝ ⠠⠁⠎⠊⠁ ⠠⠃⠥⠗⠑⠁⠥
⠠⠕⠋ ⠠⠑⠉⠕⠝⠕⠍⠊⠉ ⠠⠁⠝⠁⠇⠽⠎⠊⠎ ⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇
⠠⠉⠕⠍⠏⠁⠗⠊⠎⠕⠝ ⠠⠁⠎⠊⠁ ⠠⠃⠥⠗⠑⠁⠥ ⠠⠕⠋ ⠠⠑⠉⠕⠝⠕⠍⠊⠉
⠠⠁⠝⠁⠇⠽⠎⠊⠎ ⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇ ⠠⠉⠕⠍⠏⠁⠗⠊⠎⠕⠝ ⠠⠁⠎⠊⠁
⠠⠃⠥⠗⠑⠁⠥

⠠⠁⠇⠇⠑⠝ ⠠⠎⠞⠕⠝⠑ ⠺⠁⠎ ⠃⠕⠗⠝ ⠊⠝ ⠁ ⠎⠍⠁⠇⠇ ⠞⠕⠺⠝ ⠊⠝ ⠞⠓⠑
⠎⠕⠥⠞⠓⠑⠗⠝ ⠏⠁⠗⠞ ⠕⠋ ⠠⠉⠁⠇⠊⠋⠕⠗⠝⠊⠁. ⠠⠓⠑ ⠺⠁⠎
⠁ ⠃⠕⠕⠅⠺⠕⠗⠍. ⠠⠓⠑ ⠎⠞⠕⠗⠑ ⠺⠁⠎ ⠕⠺⠝⠑⠙ ⠃⠽ ⠠⠁⠇⠇⠑⠝
⠠⠎⠞⠕⠝⠑⠂ ⠠⠓⠊⠎ ⠋⠁⠞⠓⠑⠗⠂ ⠁⠝⠙ ⠓⠑ ⠕⠝⠇⠽ ⠺⠑⠝⠞ ⠞⠕
⠠⠠⠁⠇⠇⠑⠝⠎ ⠁⠝⠙ ⠏⠗⠕⠍⠊⠎⠑⠙ ⠓⠑ ⠺⠕⠥⠇⠙ ⠉⠁⠗⠑
⠁⠇⠺⠁⠽⠎ ⠃⠑ ⠞⠓⠑⠗⠑ ⠋⠕⠗ ⠓⠊⠍⠂ ⠃⠥⠞ ⠓⠑ ⠙⠊⠙ ⠝⠕⠞
⠎⠁⠽ ⠺⠓⠑⠝ ⠕⠗ ⠓⠕⠺ ⠞⠕ ⠉⠁⠇⠇ ⠓⠊⠍⠲

⠠⠞⠓⠑ ⠎⠞⠕⠗⠍ ⠓⠁⠙ ⠃⠗⠕⠅⠑⠝ ⠙⠕⠺⠝ ⠁⠝⠙ ⠎⠞⠕⠏⠏⠑⠙ ⠞⠓⠑
⠗⠁⠊⠝⠂ ⠞⠓⠑ ⠞⠗⠑⠑⠎ ⠺⠑⠗⠑ ⠎⠞⠊⠗⠗⠑⠙ ⠁⠝⠙ ⠎⠺⠁⠽⠑⠙⠂ ⠁⠝⠙ ⠁
⠍⠁⠝ ⠺⠓⠕ ⠓⠁⠙ ⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠓⠊⠍⠎⠑⠇⠋ ⠞⠕ ⠓⠊⠎ ⠺⠕⠗⠅ ⠊⠝
⠠⠠⠁⠇⠇⠑⠝⠎ ⠁⠝⠙ ⠏⠗⠕⠍⠊⠎⠑⠙ ⠞⠕ ⠃⠑ ⠞⠓⠑⠗⠑ ⠋⠕⠗ ⠓⠊⠍⠲

⠠⠠⠠

⠠⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠⠠

⠠⠠
⠠⠠⠠

⠠⠞⠓⠑ ⠠⠞⠓⠑ ⠺ ⠠⠙⠑ ⠠⠍⠁⠝ ⠺⠓⠕
⠇⠊⠧⠑⠙ ⠊⠝ ⠓⠊⠎ ⠠⠕⠺⠝ ⠠⠺⠕⠗⠇⠙ ⠞⠕
⠠⠞⠓⠑ ⠠⠺ ⠞⠕ ⠠⠙⠑⠎⠉⠗⠊⠃⠑ ⠠⠞⠓⠑ ⠠⠺⠕⠗⠇⠙
⠠⠍⠁⠝⠊⠋⠑⠎⠞ ⠠⠊⠝ ⠠⠞⠓⠑ ⠠⠃⠑⠛⠊⠝⠝⠊⠝⠛

⠠⠃⠽ ⠠⠙⠁⠧⠊⠙ ⠠⠎⠞⠁⠗ ⠠⠚⠕⠗⠙⠁⠝

⠠⠊ ⠠⠓⠁⠧⠑ ⠠⠃⠑⠑⠝ ⠠⠁⠎⠅⠑⠙ ⠠⠞⠕ ⠠⠞⠁⠅⠑ ⠠⠙⠊⠗⠑⠉⠞⠊⠕⠝
⠎⠞⠗⠁⠞⠑⠛⠊⠉ ⠠⠍⠁⠝⠊⠋⠑⠎⠞⠁⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠺⠕⠗⠇⠙
⠠⠞⠓⠁⠞ ⠠⠎⠓⠁⠇⠇ ⠠⠃⠑ ⠠⠞⠓⠑ ⠠⠎⠁⠍⠑ ⠃ ⠊⠝ ⠠⠞⠓⠑
⠠⠞⠓⠊⠗⠙ ⠠⠍⠕⠝⠞⠓ ⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠍⠑⠗⠉⠊⠁⠇ ⠠⠙⠊⠧⠊⠙⠑⠝⠙
⠕⠋ ⠠⠞⠓⠑ ⠠⠽⠑⠁⠗ ⠠⠺⠓⠑⠝ ⠠⠞⠓⠑ ⠠⠎⠞⠁⠞⠑ ⠠⠕⠋ ⠠⠊⠝⠙⠊⠁⠝⠁
⠠⠺⠁⠎ ⠠⠞⠓⠑ ⠠⠍⠁⠝⠊⠋⠑⠎⠞⠁⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠍⠑⠗⠉⠊⠁⠇
⠠⠙⠊⠧⠊⠙⠑⠝⠙ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠙⠊⠧⠊⠙⠑⠝⠙
⠠⠙⠑⠉⠇⠁⠗⠑⠙ ⠠⠞⠓⠁⠞ ⠠⠞⠓⠑ ⠠⠎⠁⠍⠑ ⠠⠙⠁⠽ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠍⠑⠗⠉⠊⠁⠇
⠠⠙⠊⠧⠊⠙⠑⠝⠙ ⠠⠺⠁⠎ ⠠⠏⠁⠊⠙ ⠠⠞⠕ ⠠⠞⠓⠑ ⠠⠎⠞⠁⠞⠑ ⠠⠕⠋ ⠠⠊⠝⠙⠊⠁⠝⠁
⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠍⠁⠝⠊⠋⠑⠎⠞⠁⠞⠊⠕⠝ ⠠⠕⠋

⠠⠁ ⠠⠏⠁⠛⠑ ⠕⠋ ⠠⠃⠗⠁⠊⠇⠇⠑

⠠⠏⠁⠛⠑ ⠃⠗⠁⠊⠇⠇⠑, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠁, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠁⠁, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠃, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠉, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠙, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠑, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠋, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠛, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠓, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠊, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠁⠚, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠅, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠇, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠍, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠝, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠕, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠏, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠟, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠗, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠎, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠞, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠥, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠧, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠺, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠭, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠽, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠵, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠡, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠩, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠹, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠱, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠣, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠜, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠮, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠿, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠫, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠻, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠲, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠶, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠦, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠴, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠁, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠃, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠉, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠙, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠑, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠋, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠛, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠓, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠊, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠚, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠅, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠇, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠍, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠝, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠕, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠏, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠟, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠗, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠎, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠞, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠥, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠧, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠺, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠭, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠽, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠵, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠡, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠩, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠹, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠱, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠣, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠜, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠮, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠿, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠫, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠻, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠲, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠶, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠦, ⠠⠃⠗⠁⠊⠇⠇⠑ ⠠⠠⠴.

٢٠ ٢٠٢٢٢٢

٢ ٢٢٢٢٢٢٢٢٢ ٢٢ ٢٢٢٢٢٢٢٢٢ ٢٢٢٢٢٢٢٢٢. ٢ ٢٢٢٢٢٢٢٢٢

[illegible]

[illegible]

[illegible]

⠠⠠⠀⠀⠀

الأحكام الشرعية التي يدور عليها الحكم في هذه الدعوى هي أحكام الشريعة الإسلامية في الأحوال الشخصية، فقد نصت المادة ٢٨ من لائحة ترتيب المحاكم الشرعية على أن تطبق المحاكم الشرعية في القضايا التي تدخل في اختصاصها المحلي والنوعي أحكام الشريعة الإسلامية في الأحوال الشخصية على المسلمين إلا ما استثني بقانون خاص. وقد صدر القانون رقم ٢٥ لسنة ١٩٢٩ بتعديل بعض أحكام الأحوال الشخصية، ولكنه لم يتناول بالتعديل أحكام الطلاق، فبقيت هذه الأحكام على ما كانت عليه في الشريعة الإسلامية.

وقد اختلف الفقهاء في حكم الطلاق الثلاث بلفظ واحد، فذهب جمهور الفقهاء من المذاهب الأربعة إلى أنه يقع ثلاثا، وذهب فريق آخر إلى أنه لا يقع إلا واحدة. والأصل في هذا الخلاف أن الطلاق المشروع هو الطلاق الرجعي، وهو الذي يملك فيه الزوج مراجعة زوجته في العدة دون حاجة إلى عقد جديد. أما الطلاق الثلاث فهو طلاق بائن بينونة كبرى، لا تحل المطلقة بعده لمطلقها إلا بعقد جديد ومهر جديد بعد انقضاء عدتها من زوج آخر.

⠠⠁ ⠠⠠⠠⠁ ⠠⠃ ⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠⠁
⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁
⠠⠠⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠⠁
⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁
⠠⠠⠠⠠⠠⠁
⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁
⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁
⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁
⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁ ⠠⠠⠠⠠⠠⠁

[illegible]

[illegible]

[illegible]

⠀⠀⠀

⠠⠠
⠠⠠⠠

⠠ ⠠⠏⠗⠕⠉⠑⠎⠎ ⠠⠁⠝⠁⠇⠽⠎⠊⠎ ⠠⠁⠝⠁⠇⠽⠎⠊⠎ ⠠⠁⠝⠁⠇⠽⠎⠊

⠠⠞⠓⠑ ⠠⠠⠓⠑⠁⠙⠊⠝⠛⠎⠠⠠ ⠏⠁⠛⠑ ⠺⠁⠎ ⠞⠓⠑ ⠏⠁⠛⠑ ⠁⠞ ⠺⠓⠊⠉⠓ ⠁ ⠏⠑⠗⠎⠕⠝ ⠺⠁⠎ ⠇⠊⠅⠑⠇⠽ ⠞⠕ ⠎⠞⠁⠗⠞ ⠗⠑⠁⠙⠊⠝⠛ ⠁ ⠃⠕⠕⠅ ⠋⠗⠕⠍ ⠞⠓⠑ ⠃⠑⠛⠊⠝⠝⠊⠝⠛ ⠕⠋ ⠞⠓⠑ ⠋⠊⠗⠎⠞ ⠉⠓⠁⠏⠞⠑⠗ ⠕⠋ ⠞⠓⠑ ⠋⠊⠗⠎⠞ ⠃⠕⠕⠅ ⠕⠋ ⠞⠓⠑ ⠠⠠⠓⠑⠁⠙⠊⠝⠛⠎⠠⠠ ⠎⠑⠗⠊⠑⠎ ⠺⠁⠎ ⠁ ⠙⠑⠇⠊⠛⠓⠞⠋⠥⠇ ⠎⠥⠗⠏⠗⠊⠎⠑ ⠞⠕ ⠍⠑⠁⠝⠽ ⠕⠋ ⠍⠽ ⠋⠗⠊⠑⠝⠙⠎ ⠺⠓⠕ ⠓⠁⠙ ⠝⠑⠧⠑⠗ ⠗⠑⠁⠙ ⠁ ⠃⠕⠕⠅ ⠕⠗ ⠎⠑⠑⠝ ⠁ ⠍⠕⠧⠊⠑ ⠃⠑⠋⠕⠗⠑ ⠞⠓⠑⠽ ⠺⠑⠗⠑ ⠁⠇⠇ ⠁⠃⠕⠥⠞ ⠞⠕ ⠞⠁⠅⠑ ⠁ ⠞⠗⠊⠏ ⠞⠕ ⠞⠓⠑ ⠍⠕⠥⠝⠞⠁⠊⠝⠎ ⠁⠝⠙ ⠺⠁⠎ ⠝⠑⠧⠑⠗ ⠁⠛⠁⠊⠝ ⠎⠑⠑⠝ ⠁⠛⠁⠊⠝ ⠁⠋⠞⠑⠗ ⠞⠓⠁⠞ ⠙⠁⠽ ⠁⠝⠙ ⠊ ⠝⠑⠧⠑⠗ ⠅⠝⠑⠺ ⠺⠓⠁⠞ ⠓⠁⠏⠏⠑⠝⠑⠙ ⠞⠕ ⠞⠓⠁⠞ ⠏⠁⠛⠑ ⠁⠛⠁⠊⠝ ⠁⠝⠙ ⠊ ⠝⠑⠧⠑⠗ ⠅⠝⠕

⠠⠁⠞ ⠞⠓⠑ ⠉⠇⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠍⠑⠑⠞⠊⠝⠛ ⠓⠑ ⠞⠕⠕⠅ ⠞⠓⠑ ⠉⠓⠁⠊⠗ ⠁⠝⠙ ⠎⠁⠊⠙ ⠊⠞ ⠺⠁⠎ ⠍⠽ ⠏⠇⠑⠁⠎⠥⠗⠑ ⠞⠕ ⠏⠗⠑⠎⠊⠙⠑ ⠁⠞ ⠞⠓⠊⠎ ⠍⠑⠑⠞⠊⠝⠛ ⠠⠁⠝⠙ ⠊ ⠁⠍ ⠎⠥⠗⠑ ⠞⠓⠁⠞ ⠁⠇⠇ ⠞⠓⠑ ⠍⠑⠍⠃⠑⠗⠎ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠺⠊⠇⠇ ⠗⠑⠍⠁⠊⠝ ⠊⠝ ⠞⠓⠑ ⠗⠕⠕⠍ ⠁⠋⠞⠑⠗ ⠞⠓⠑ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠍⠑⠑⠞⠎ ⠁⠝⠙ ⠞⠓⠁⠞ ⠊⠞ ⠺⠊⠇⠇ ⠃⠑ ⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠗⠕⠙⠥⠉⠞⠊⠧⠑ ⠍⠑⠑⠞⠊⠝⠛ ⠕⠋ ⠞⠓⠑ ⠽⠑⠁⠗ ⠠⠁⠝⠙ ⠊ ⠁⠍ ⠎⠥⠗⠑ ⠞⠓⠁⠞ ⠁⠇⠇ ⠞⠓⠑ ⠍⠑⠍⠃⠑⠗⠎ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠺⠊⠇⠇ ⠗⠑⠍⠁⠊⠝ ⠊⠝ ⠞⠓⠑ ⠗⠕⠕⠍ ⠁⠋⠞⠑⠗ ⠞⠓⠑ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠍⠑⠑⠞⠎ ⠁⠝⠙ ⠞⠓⠁⠞ ⠊⠞ ⠺⠊⠇⠇ ⠃⠑ ⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠗⠕⠙⠥⠉⠞⠊⠧⠑ ⠍⠑⠑⠞⠊⠝⠛ ⠕⠋ ⠞⠓⠑ ⠽⠑⠁⠗

⠠⠁⠞ ⠞⠓⠑ ⠉⠇⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠍⠑⠑⠞⠊⠝⠛ ⠓⠑ ⠞⠕⠕⠅ ⠞⠓⠑ ⠉⠓⠁⠊⠗ ⠁⠝⠙ ⠎⠁⠊⠙ ⠊⠞ ⠺⠁⠎ ⠍⠽ ⠏⠇⠑⠁⠎⠥⠗⠑ ⠞⠕ ⠏⠗⠑⠎⠊⠙⠑ ⠁⠞ ⠞⠓⠊⠎ ⠍⠑⠑⠞⠊⠝⠛ ⠠⠁⠝⠙ ⠊ ⠁⠍ ⠎⠥⠗⠑ ⠞⠓⠁⠞ ⠁⠇⠇ ⠞⠓⠑ ⠍⠑⠍⠃⠑⠗⠎ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠺⠊⠇⠇ ⠗⠑⠍⠁⠊⠝ ⠊⠝ ⠞⠓⠑ ⠗⠕⠕⠍ ⠁⠋⠞⠑⠗ ⠞⠓⠑ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠍⠑⠑⠞⠎ ⠁⠝⠙ ⠞⠓⠁⠞ ⠊⠞ ⠺⠊⠇⠇ ⠃⠑ ⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠗⠕⠙⠥⠉⠞⠊⠧⠑ ⠍⠑⠑⠞⠊⠝⠛ ⠕⠋ ⠞⠓⠑ ⠽⠑⠁⠗

⠠⠃
⠠⠠⠠

⠠⠍⠁⠗⠞⠁ ⠠⠍⠁⠗⠊⠁ ⠠⠉⠁⠍⠁⠗⠛⠕ ⠠⠃⠁⠗⠗⠊⠕⠎ ⠠⠎⠁⠝ ⠠⠍⠊⠛⠥⠑⠇ ⠝
⠠⠁⠝⠛⠑⠇ ⠙⠑ ⠠⠍⠁⠗⠞⠁ ⠙⠑ ⠠⠍⠁⠗⠊⠁ ⠠⠇⠁⠎ ⠠⠍⠑⠙⠁⠎ ⠠⠝⠕
⠝ ⠠⠙⠊⠑⠠⠛⠕ ⠙ ⠠⠉⠁⠍⠁⠗⠛⠕ ⠠⠃⠁⠗⠗⠊⠕⠎ ⠠⠇⠁⠎ ⠠⠍⠑⠙⠁⠎
⠠⠙⠑ ⠠⠍⠁⠗⠞⠁ ⠠⠁⠝⠙⠗⠑⠁ ⠠⠁⠇⠑⠚⠁⠝⠙⠗⠁ ⠠⠁⠇⠑⠚⠁⠝⠙⠗⠁

⠈⠁ ⠀⠀⠀⠀⠀⠠⠞⠓⠑ ⠞⠗⠥⠞⠓ ⠁⠃⠕⠥⠞ ⠞⠓⠑ ⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠃⠕⠙⠽ ⠕⠋ ⠉⠓⠗⠊⠎⠞ ⠀⠀⠀⠀⠀

[illegible]

[illegible] [illegible], [illegible]. [illegible] [illegible]

[illegible] [illegible]

[illegible] [illegible] [illegible] [

[illegible] [illegible] [illegible] [

⠠⠓ ⠉⠗⠑⠁⠞⠑ ⠁ ⠝⠑⠺ ⠃⠗⠁⠝⠉⠓ ⠁⠞ ⠁ ⠎⠏⠑⠉⠊⠋⠊⠉ ⠏⠕⠊⠝⠞ ⠊⠝ ⠞⠓⠑ ⠓⠊⠎⠞⠕⠗⠽ ⠕⠋ ⠞⠓⠑ ⠗⠑⠏⠕⠎⠊⠞⠕⠗⠽.
⠠⠞⠓⠊⠎ ⠉⠕⠍⠍⠁⠝⠙ ⠉⠗⠑⠁⠞⠑⠎ ⠁ ⠝⠑⠺ ⠗⠑⠏⠕ ⠁⠞ ⠞⠓⠑ ⠎⠏⠑⠉⠊⠋⠊⠑⠙ ⠏⠁⠞⠓ ⠁⠝⠙ ⠁⠙⠙⠎ ⠁ ⠉⠕⠍⠍⠊⠞ ⠞⠕ ⠞⠓⠁⠞ ⠃⠗⠁⠝⠉⠓.
⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇⠇⠽, ⠞⠓⠑ ⠉⠕⠍⠍⠁⠝⠙ ⠁⠇⠎⠕ ⠉⠗⠑⠁⠞⠑⠎ ⠁ ⠝⠑⠺ ⠗⠑⠏⠕ ⠁⠞ ⠞⠓⠑ ⠎⠏⠑⠉⠊⠋⠊⠑⠙ ⠏⠁⠞⠓ ⠁⠝⠙ ⠁⠙⠙⠎ ⠁ ⠉⠕⠍⠍⠊⠞ ⠞⠕ ⠞⠓⠁⠞ ⠃⠗⠁⠝⠉⠓.
⠠⠞⠓⠑ ⠉⠕⠍⠍⠁⠝⠙ ⠁⠇⠎⠕ ⠉⠗⠑⠁⠞⠑⠎ ⠁ ⠝⠑⠺ ⠗⠑⠏⠕ ⠁⠞ ⠞⠓⠑ ⠎⠏⠑⠉⠊⠋⠊⠑⠙ ⠏⠁⠞⠓ ⠁⠝⠙ ⠁⠙⠙⠎ ⠁ ⠉⠕⠍⠍⠊⠞ ⠞⠕ ⠞⠓⠁⠞ ⠃⠗⠁⠝⠉⠓.
⠠⠞⠓⠑ ⠉⠕⠍⠍⠁⠝⠙ ⠁⠇⠎⠕ ⠉⠗⠑⠁⠞⠑⠎ ⠁ ⠝⠑⠺ ⠗⠑⠏⠕ ⠁⠞ ⠞⠓⠑ ⠎⠏⠑⠉⠊⠋⠊⠑⠙ ⠏⠁⠞⠓ ⠁⠝⠙ ⠁⠙⠙⠎ ⠁ ⠉⠕⠍⠍⠊⠞ ⠞⠕ ⠞⠓⠁⠞ ⠃⠗⠁⠝⠉⠓.
⠠⠞⠓⠑ ⠉⠕⠍⠍⠁⠝⠙ ⠁⠇⠎⠕ ⠉⠗⠑⠁⠞⠑⠎ ⠁ ⠝⠑⠺ ⠗⠑⠏⠕ ⠁⠞ ⠞⠓⠑ ⠎⠏⠑⠉⠊⠋⠊⠑⠙ ⠏⠁⠞⠓ ⠁⠝⠙ ⠁⠙⠙⠎ ⠁ ⠉⠕⠍⠍⠊⠞ ⠞⠕ ⠞⠓⠁⠞ ⠃⠗⠁⠝⠉⠓.
⠠⠞⠓⠑ ⠉⠕⠍⠍⠁⠝⠙ ⠁⠇⠎⠕ ⠉⠗⠑⠁⠞⠑⠎ ⠁ ⠝⠑⠺ ⠗⠑⠏⠕ ⠁⠞ ⠞⠓⠑ ⠎⠏⠑⠉⠊⠋⠊⠑⠙ ⠏⠁⠞⠓ ⠁⠝⠙ ⠁⠙⠙⠎ ⠁ ⠉⠕⠍⠍⠊⠞ ⠞⠕ ⠞⠓⠁⠞ ⠃⠗⠁⠝⠉⠓.
⠠⠞⠓⠑ ⠉⠕⠍⠍⠁⠝⠙ ⠁⠇⠎⠕ ⠉⠗⠑⠁⠞⠑⠎ ⠁ ⠝⠑⠺ ⠗⠑⠏⠕ ⠁⠞ ⠞⠓⠑ ⠎⠏⠑⠉⠊⠋⠊⠑⠙ ⠏⠁⠞⠓ ⠁⠝⠙ ⠁⠙⠙⠎ ⠁ ⠉⠕⠍⠍⠊⠞ ⠞⠕ ⠞⠓⠁⠞ ⠃⠗⠁⠝⠉⠓.

[illegible]

[illegible]

[illegible]

⠠⠞⠓⠑ ⠞⠗⠁⠙⠑ ⠕⠋ ⠠⠺⠕⠗⠙⠎ ⠁⠝⠙ ⠠⠏⠓⠗⠁⠎⠑⠎ ⠞⠓⠁⠞ ⠺⠑⠗⠑ ⠝⠕⠞ ⠞⠓⠑
⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠕⠥⠗ ⠝⠑⠊⠛⠓⠃⠕⠗⠎ ⠺⠁⠎ ⠝⠕⠞ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋
⠞⠓⠑ ⠠⠏⠓⠁⠗⠊⠎⠑⠑⠎⠲ ⠠⠓⠑ ⠺⠁⠎ ⠝⠕⠞ ⠁ ⠠⠏⠓⠁⠗⠊⠎⠑⠑⠲ ⠠⠓⠑ ⠺⠁⠎ ⠁ ⠠⠞⠁⠭
⠉⠕⠇⠇⠑⠉⠞⠕⠗⠲ ⠠⠓⠑ ⠺⠁⠎ ⠁ ⠠⠞⠁⠭ ⠉⠕⠇⠇⠑⠉⠞⠕⠗⠲ ⠠⠓⠑ ⠺⠁⠎ ⠁ ⠠⠞⠁⠭
⠉⠕⠇⠇⠑⠉⠞⠕⠗⠲ ⠠⠓⠑ ⠺⠁⠎ ⠁ ⠠⠞⠁⠭ ⠉⠕⠇⠇⠑⠉⠞⠕⠗⠲

⠠⠞⠓⠑ ⠠⠏⠗⠊⠝⠉⠑ ⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠏⠁⠓⠁⠝

⠠⠞⠓⠑ ⠠⠏⠗⠊⠝⠉⠑ ⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠏⠁⠓⠁⠝ ⠍

⠠⠃⠁⠎⠞⠁⠗⠙ ⠠⠠⠃⠥⠇⠇ ⠠⠞⠗⠁⠝⠎⠏⠕⠗⠞⠁⠞⠊⠕⠝ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠠⠁⠉⠞ ⠠⠁⠎
⠠⠠⠁⠍⠑⠝⠙⠑⠙ ⠠⠠⠁⠍⠑⠝⠙⠍⠑⠝⠞⠎ ⠠⠠⠁⠍⠑⠝⠙⠍⠑⠝⠞⠎ ⠠⠠⠁⠍⠑⠝⠙⠍⠑⠝⠞⠎ ⠠⠠⠁⠍⠑⠝⠙⠍⠑⠝⠞⠎

⠠⠠
⠠⠠⠠

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

⠠⠠⠓⠕⠺ ⠓⠁⠎ ⠞⠓⠑ ⠠⠠⠍⠁⠗⠅⠑⠞⠎⠠⠠ ⠉⠓⠁⠝⠛⠑⠙ ⠞⠓⠑ ⠺⠁⠽ ⠏⠑⠕⠏⠇⠑ ⠎⠓⠕⠏ ⠠⠃⠽ ⠠⠠⠓⠕⠺ ⠞⠓⠑⠽ ⠁⠗⠑ ⠍⠁⠙⠑⠠⠠ ⠞⠕ ⠞⠓⠑ ⠠⠠⠓⠕⠺ ⠞⠓⠑⠽ ⠁⠗⠑⠠⠠ ⠞⠕ ⠞⠓⠑ ⠠⠠⠍⠁⠗⠅⠑⠞⠎⠠⠠ ⠞⠓⠑⠍⠎⠑⠇⠧⠑⠎

⠠ ⠞⠓⠑⠗⠑⠋⠕⠗⠑ ⠔ ⠃⠁⠎⠊⠍⠏⠕⠎⠎⠊⠃⠇⠑ ⠠ ⠉⠁⠎⠑ ⠊⠝⠧⠕⠇⠧⠊⠝⠛ ⠠⠙⠊⠅ ⠁ ⠞⠁⠅⠑
⠎⠁⠽⠎ ⠺⠓⠑⠝ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠔ ⠉⠕⠍⠏⠁⠗⠁⠃⠇⠑ ⠞⠕ ⠙⠊⠅ ⠠⠃⠥⠞ ⠠⠙⠊⠅
⠔ ⠞⠓⠑ ⠉⠁⠎⠑ ⠊⠞ ⠺⠁⠎ ⠙⠕⠝⠑ ⠞⠓⠑ ⠕⠞⠓⠑⠗ ⠺⠁⠽ ⠁⠗⠕⠥⠝⠙
⠁⠝⠙ ⠁ ⠛⠕⠕⠙ ⠙⠑⠁⠇ ⠕⠋ ⠍⠕⠝⠑⠽ ⠺⠁⠎ ⠎⠁⠧⠑⠙ ⠕⠝ ⠞⠓⠑
⠠ ⠉⠕⠍⠏⠁⠝⠽ ⠞⠕⠕ ⠠⠙⠊⠅

⠠⠠ ⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑ ⠠⠊ ⠠⠠
⠠⠠ ⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑ ⠠⠊ ⠠⠠
⠠⠠ ⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑ ⠠⠊ ⠠⠠

⠠⠠
⠠⠠⠠

⠠⠞⠓⠑ ⠞⠗⠁⠙⠑ ⠍⠁⠗⠅ ⠁⠝⠙ ⠎⠑⠗⠧⠊⠉⠑ ⠍⠁⠗⠅ ⠁⠗⠑ ⠞⠓⠑
⠗⠑⠛⠊⠎⠞⠑⠗⠑⠙ ⠞⠗⠁⠙⠑ ⠍⠁⠗⠅⠎ ⠕⠋ ⠞⠓⠑ ⠥⠝⠊⠞⠑⠙ ⠎⠞⠁⠞⠑⠎
⠕⠋ ⠁⠍⠑⠗⠊⠉⠁ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠓⠕⠎⠑ ⠏⠗⠊⠝⠉⠊⠏⠁⠇
⠃⠥⠎⠊⠝⠑⠎⠎ ⠊⠎ ⠞⠕ ⠏⠗⠕⠧⠊⠙⠑ ⠁ ⠎⠁⠋⠑ ⠁⠝⠙ ⠎⠑⠉⠥⠗⠑
⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞ ⠋⠕⠗ ⠁⠇⠇ ⠕⠋ ⠞⠓⠑ ⠏⠑⠕⠏⠇⠑ ⠺⠓⠕
⠺⠕⠗⠅ ⠕⠝ ⠞⠓⠑ ⠏⠇⠁⠝⠞⠁⠞⠊⠕⠝⠲ ⠠⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽
⠊⠎ ⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠞⠕ ⠏⠗⠕⠧⠊⠙⠊⠝⠛ ⠁ ⠎⠁⠋⠑
⠁⠝⠙ ⠎⠑⠉⠥⠗⠑ ⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞ ⠋⠕⠗ ⠁⠇⠇ ⠕⠋ ⠞⠓⠑
⠏⠑⠕⠏⠇⠑ ⠺⠓⠕ ⠺⠕⠗⠅ ⠕⠝ ⠞⠓⠑ ⠏⠇⠁⠝⠞⠁⠞⠊⠕⠝⠲

[illegible]

[illegible]

[illegible]

⠠⠞ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠍⠊⠎⠎⠊⠕⠝⠑⠗ ⠕⠋ ⠠⠏⠥⠃⠇⠊⠉ ⠠⠺⠕⠗⠅⠎ ⠍⠁⠽ ⠉⠕⠍⠏⠁⠉⠞ ⠁ ⠠⠉⠕⠙⠑
⠕⠋ ⠠⠏⠗⠁⠉⠞⠊⠉⠑ ⠠⠏⠗⠑⠎⠉⠗⠊⠃⠊⠝⠛ ⠠⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇ ⠠⠠⠊⠝⠋⠕⠗⠍⠁⠞⠊⠕⠝ ⠕⠗ ⠊⠞ ⠍⠁⠽
⠃⠑ ⠠⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇ ⠠⠠⠊⠝⠋⠕⠗⠍⠁⠞⠊⠕⠝ ⠞⠕ ⠉⠕⠍⠏⠇⠽ ⠺⠊⠞⠓

⠠⠠⠁ ⠠⠠⠏⠥⠃⠇⠊⠉ ⠠⠠⠺⠕⠗⠅ ⠍⠁⠽ ⠃⠑ ⠠⠠⠙⠑⠋⠊⠝⠑⠙ ⠁⠎ ⠁ ⠠⠠⠉⠕⠍⠏⠁⠉⠞
⠠⠠⠏⠗⠁⠉⠞⠊⠉⠑ ⠠⠠⠏⠗⠑⠎⠉⠗⠊⠃⠊⠝⠛ ⠠⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇ ⠠⠠⠊⠝⠋⠕⠗⠍⠁⠞⠊⠕⠝
⠕⠗ ⠠⠠⠊⠝ ⠠⠠⠁ ⠠⠠⠉⠕⠍⠏⠁⠉⠞ ⠠⠠⠏⠗⠁⠉⠞⠊⠉⠑ ⠠⠠⠏⠗⠑⠎⠉⠗⠊⠃⠊⠝⠛
⠠⠠⠁⠙⠙⠊⠞⠊⠕⠝⠁⠇ ⠠⠠⠊⠝⠋⠕⠗⠍⠁⠞⠊⠕⠝ ⠠⠠⠁⠎ ⠠⠠⠙⠑⠋⠊⠝⠑⠙ ⠠⠠⠃⠑⠇⠕⠺⠂

⠠⠞⠓⠑ ⠠⠠⠠ ⠠⠠⠠⠠⠠

⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇ ⠁⠊⠗⠏⠕⠗⠞ ⠁⠉⠞⠊⠧⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠕⠋ ⠞⠓⠑ ⠍⠊⠙⠙⠇⠑
⠞⠕⠏⠁⠝ ⠗⠑⠏⠥⠃⠇⠊⠉ ⠺⠊⠞⠓ ⠊⠞⠎ ⠓⠑⠁⠙⠟⠥⠁⠗⠞⠑⠗⠎ ⠁⠞ ⠞⠓⠑
⠍⠁⠗⠞⠊⠝ ⠉⠑⠝⠞⠑⠗ ⠕⠋ ⠞⠓⠑ ⠁⠊⠗⠏⠕⠗⠞ ⠁⠝⠙ ⠊⠎ ⠞⠓⠑ ⠍⠁⠊⠝
⠓⠥⠃ ⠋⠕⠗ ⠞⠗⠁⠝⠎⠏⠕⠗⠞⠁⠞⠊⠕⠝ ⠁⠉⠗⠕⠎⠎ ⠞⠓⠑ ⠉⠕⠥⠝⠞⠗⠽ ⠁⠝⠙
⠊⠞ ⠊⠎ ⠁⠇⠎⠕ ⠁ ⠍⠁⠚⠕⠗ ⠎⠕⠥⠗⠉⠑ ⠕⠋ ⠑⠍⠏⠇⠕⠽⠍⠑⠝⠞
⠋⠕⠗ ⠍⠁⠝⠽ ⠏⠑⠕⠏⠇⠑ ⠁⠗⠕⠥⠝⠙ ⠞⠓⠑ ⠺⠕⠗⠇⠙⠲ ⠠⠊⠝ ⠞⠓⠑
⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠊⠞ ⠊⠎ ⠁ ⠍⠁⠚⠕⠗ ⠎⠕⠥⠗⠉⠑ ⠕⠋
⠑⠍⠏⠇⠕⠽⠍⠑⠝⠞ ⠋⠕⠗ ⠍⠁⠝⠽ ⠏⠑⠕⠏⠇⠑ ⠁⠗⠕⠥⠝⠙ ⠞⠓⠑
⠺⠕⠗⠇⠙ ⠺⠓⠕ ⠁⠗⠑ ⠁⠇⠎⠕ ⠎⠑⠇⠋ ⠑⠍⠏⠇⠕⠽⠑⠙ ⠁⠞ ⠁
⠏⠗⠊⠧⠁⠞⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠕⠗ ⠊⠝ ⠁ ⠍⠊⠭⠑⠙ ⠑⠉⠕⠝⠕⠍⠽
⠠⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠊⠎
⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠕⠋ ⠏⠑⠕⠏⠇⠑ ⠺⠕⠗⠅⠊⠝⠛ ⠞⠕⠛⠑⠞⠓⠑⠗
⠞⠕ ⠍⠁⠅⠑ ⠁ ⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠊⠝ ⠞⠓⠑ ⠺⠕⠗⠇⠙⠲

⠠⠓⠑ ⠞⠕⠕⠅ ⠓⠊⠎ ⠞⠁⠅⠑ ⠁⠝⠙ ⠓⠊⠎ ⠎⠞⠁⠋⠋ ⠺⠑⠝⠞ ⠕⠝ ⠁ ⠎⠑⠁⠗⠉⠓ ⠋⠕⠗ ⠁ ⠝⠑⠺ ⠓⠕⠍⠑ ⠋⠕⠗ ⠓⠊⠍⠎⠑⠇⠋ ⠁⠝⠙ ⠓⠊⠎ ⠋⠁⠍⠊⠇⠽ ⠁⠋⠞⠑⠗ ⠓⠑ ⠓⠁⠙ ⠎⠑⠞⠞⠇⠑⠙ ⠙⠕⠺⠝ ⠁⠞ ⠁ ⠎⠞⠁⠞⠊⠕⠝ ⠊⠝ ⠞⠓⠑ ⠍⠊⠙⠙⠇⠑ ⠕⠋ ⠝⠕⠺⠓⠑⠗⠑ ⠁⠝⠙ ⠓⠁⠙ ⠝⠕ ⠊⠙⠑⠁ ⠺⠓⠁⠞ ⠓⠑ ⠺⠁⠎ ⠙⠕⠊⠝⠛ ⠓⠑⠗⠑ ⠕⠗ ⠺⠓⠽ ⠓⠑ ⠺⠁⠎ ⠙⠕⠊⠝⠛ ⠞⠓⠑⠗⠑ ⠁⠞ ⠁⠇⠇⠂ ⠓⠑ ⠓⠁⠙ ⠝⠕ ⠊⠙⠑⠁ ⠺⠓⠁⠞

⠠⠠
⠠⠠⠠

⠠⠞⠓⠑ ⠠⠉⠕⠍⠍⠊⠎⠎⠊⠕⠝⠑⠗ ⠍⠁⠽ ⠁⠎⠅ ⠋⠕⠗ ⠁ ⠉⠕⠏⠽ ⠕⠋ ⠁ ⠗⠑⠉⠕⠗⠙ ⠕⠋ ⠁ ⠙⠑⠁⠞⠓ ⠕⠗ ⠊⠝⠚⠥⠗⠽ ⠊⠝ ⠁ ⠉⠁⠎⠑ ⠏⠑⠝⠙⠊⠝⠛ ⠁⠞ ⠞⠓⠑ ⠞⠊⠍⠑

⠠⠞⠓⠑ ⠠⠍⠁⠝ ⠺⠓⠕ ⠠⠅⠝⠑⠺ ⠠⠞⠓⠑ ⠠⠁⠗⠉⠓ ⠍⠁⠉⠓⠊⠝⠑

⠠⠞⠓⠑ ⠠⠍⠁⠝ ⠺⠓⠕ ⠠⠅⠝⠑⠺ ⠠⠞⠓⠑ ⠠⠁⠗⠉⠓ ⠍⠁⠉⠓⠊⠝⠑ ⠠⠺⠁⠎ ⠠⠁ ⠠⠍⠁⠝ ⠠

⠠⠠
⠠⠠⠠

⠠⠞⠓⠑ ⠠⠍⠁⠝ ⠺⠓⠕ ⠠⠺⠁⠎ ⠠⠞⠕⠕ ⠠⠎⠍⠁⠇⠇

⠠⠞⠓⠑ ⠠⠍⠁⠝ ⠺⠓⠕ ⠠⠺⠁⠎ ⠠⠞⠕⠕ ⠠⠎⠍⠁⠇⠇

⠠⠞⠓⠑ ⠠⠍⠁⠝ ⠺⠓⠕ ⠠⠺⠁⠎ ⠠⠞⠕⠕ ⠠⠎⠍⠁⠇⠇ ⠠⠊⠝ ⠠⠓⠊⠎
⠠⠍⠊⠝⠙ ⠠⠞⠓⠑ ⠠⠏⠗⠕⠃⠇⠑⠍⠎ ⠠⠕⠋ ⠠⠍⠁⠝⠅⠊⠝⠙ ⠠⠊⠎⠠⠄
⠠⠊⠞ ⠠⠓⠁⠙ ⠁ ⠠⠍⠕⠗⠁⠇ ⠠⠙⠊⠍⠑⠝⠎⠊⠕⠝⠠⠄ ⠠⠞⠓⠑ ⠠⠏⠗⠕⠃⠇⠑⠍
⠕⠋ ⠠⠅⠝⠕⠺⠊⠝⠛ ⠠⠺⠓⠁⠞ ⠠⠊⠎ ⠠⠗⠊⠛⠓⠞ ⠠⠁⠝⠙ ⠠⠺⠗⠕⠝⠛⠠⠄
⠠⠞⠓⠑ ⠠⠏⠗⠕⠃⠇⠑⠍ ⠠⠊⠎ ⠠⠞⠓⠁⠞ ⠠⠺⠑ ⠠⠙⠕ ⠠⠝⠕⠞ ⠠⠅⠝⠕⠺
⠠⠺⠓⠁⠞ ⠠⠺⠑ ⠠⠙⠕ ⠠⠅⠝⠕⠺⠠⠄ ⠠⠺⠑ ⠠⠙⠕ ⠠⠝⠕⠞ ⠠⠅⠝⠕⠺ ⠠⠺⠓⠁⠞
⠠⠺⠑ ⠠⠙⠕ ⠠⠅⠝⠕⠺⠠⠄ ⠠⠺⠑ ⠠⠙⠕ ⠠⠝⠕⠞ ⠠⠅⠝⠕⠺ ⠠⠺⠓⠁⠞ ⠠⠺⠑
⠠⠙⠕ ⠠⠅⠝⠕⠺⠠⠄ ⠠⠺⠑ ⠠⠙⠕ ⠠⠝⠕⠞ ⠠⠅⠝⠕⠺ ⠠⠺⠓⠁⠞ ⠠⠺⠑ ⠠⠙⠕
⠠⠅⠝⠕⠺⠠⠄ ⠠⠺⠑ ⠠⠙⠕ ⠠⠝⠕⠞ ⠠⠅⠝⠕⠺ ⠠⠺⠓⠁⠞ ⠠⠺⠑ ⠠⠙⠕ ⠠⠅⠝⠕⠺
⠠⠺⠓⠁⠞ ⠠⠺⠑ ⠠⠙⠕ ⠠⠅⠝⠕⠺⠠⠄ ⠠⠺⠑ ⠠⠙⠕ ⠠⠝⠕⠞ ⠠⠅⠝⠕⠺ ⠠⠺⠓⠁⠞
⠠⠺⠑

⠠ ⠠⠎⠑⠉ ⠠⠏⠗⠕⠏⠑⠗ ⠠⠃⠁⠎⠊⠎ ⠠⠕⠋ ⠠⠁ ⠠⠍⠁⠝⠁⠛⠑⠗⠎ ⠠⠞⠕⠏ ⠠⠍⠁⠝⠁⠛⠑⠗⠎ ⠠⠁⠝⠙
⠠⠁⠎ ⠠⠁ ⠠⠎⠑⠞ ⠠⠕⠋ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠕⠗ ⠠⠁ ⠠⠏⠁⠗⠞ ⠠⠕⠋ ⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠽
⠠⠕⠗⠛⠁⠝⠊⠵⠁⠞⠊⠕⠝ ⠠⠎⠽⠎⠞⠑⠍⠁⠞⠊⠉⠁⠇⠇⠽ ⠠⠁ ⠠⠇⠁⠗⠛⠑⠞⠁⠃⠇⠑ ⠠⠝⠥⠍⠃⠑⠗ ⠠⠕⠋ ⠠⠏⠑⠕⠏⠇⠑
⠠⠺⠕⠗⠅⠊⠝⠛ ⠠⠕⠝ ⠠⠁ ⠠⠉⠕⠍⠍⠕⠝ ⠠⠞⠁⠎⠅ ⠠⠁⠝⠙ ⠠⠏⠗⠕⠙⠥⠉⠑⠎ ⠠⠁ ⠠⠉⠕⠍⠍⠕⠙⠊⠞⠽
⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠥⠎⠑⠋⠥⠇ ⠠⠋⠕⠗ ⠠⠁ ⠠⠉⠕⠍⠏⠥⠞⠁⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠽
⠠⠕⠗ ⠠⠁ ⠠⠏⠁⠗⠞ ⠠⠕⠋ ⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠥⠎⠑⠋⠥⠇ ⠠⠋⠕⠗
⠠⠁ ⠠⠉⠕⠍⠏⠥⠞⠁⠞⠊⠕⠝ ⠠⠕⠗ ⠠⠁ ⠠⠏⠁⠗⠞ ⠠⠕⠋ ⠠⠁ ⠠⠉⠕⠍⠏⠥⠞⠁⠞⠊⠕⠝ ⠠⠞⠓⠁⠞
⠠⠊⠎ ⠠⠥⠎⠑⠋⠥⠇ ⠠⠋⠕⠗ ⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠕⠗ ⠠⠁ ⠠⠏⠁⠗⠞ ⠠⠕⠋ ⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠽
⠠⠕⠗⠛⠁⠝⠊⠵⠁⠞⠊⠕⠝ ⠠⠎⠽⠎⠞⠑⠍⠁⠞⠊⠉⠁⠇⠇⠽ ⠠⠁ ⠠⠇⠁⠗⠛⠑⠞⠁⠃⠇⠑ ⠠⠝⠥⠍⠃⠑⠗ ⠠⠕⠋
⠠⠏⠑⠕⠏⠇⠑ ⠠⠺⠕⠗⠅⠊⠝⠛ ⠠⠕⠝ ⠠⠁ ⠠⠉⠕⠍⠍⠕⠝ ⠠⠞⠁⠎⠅ ⠠⠁⠝⠙ ⠠⠏⠗⠕⠙⠥⠉⠑⠎ ⠠⠁
⠠⠉⠕⠍⠍⠕⠙⠊⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠥⠎⠑⠋⠥⠇ ⠠⠋⠕⠗ ⠠⠁ ⠠⠉⠕⠍⠏⠥⠞⠁⠞⠊⠕⠝ ⠠⠕⠗
⠠⠁ ⠠⠏⠁⠗⠞ ⠠⠕⠋ ⠠⠁ ⠠⠉⠕⠍⠏⠥⠞⠁⠞⠊⠕⠝ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠥⠎⠑⠋⠥⠇ ⠠⠋⠕⠗ ⠠⠁
⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠕⠗⠛⠁⠝⠊⠵⠁⠞⠊⠕⠝ ⠠⠎⠽⠎⠞⠑⠍⠁⠞⠊⠉⠁⠇⠇⠽ ⠠⠁ ⠠⠇⠁⠗⠛⠑⠞⠁⠃⠇⠑
⠠⠝⠥⠍⠃⠑⠗ ⠠⠕⠋ ⠠⠏⠑⠕⠏⠇⠑ ⠠⠺⠕⠗⠅⠊⠝⠛ ⠠⠕⠝ ⠠⠁ ⠠⠉⠕⠍⠍⠕⠝ ⠠⠞⠁⠎⠅ ⠠⠁⠝⠙
⠠⠏⠗⠕⠙⠥⠉⠑⠎ ⠠⠁ ⠠⠉⠕⠍⠍⠕⠙⠊⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠥⠎⠑⠋⠥⠇ ⠠⠋⠕⠗ ⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠽
⠠⠕⠗ ⠠⠁ ⠠⠏⠁⠗⠞ ⠠⠕⠋ ⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠽

à à à

⠠⠠⠠

⠠⠇⠊⠝⠅ ⠠⠠⠎⠏⠁⠝⠊⠎⠓ ⠠⠠⠏⠕⠏⠥⠇⠁⠗ ⠠⠠⠎⠕⠝⠛⠎ ⠠⠠⠁⠗⠕⠥⠝⠙ ⠠⠠⠍⠑ ⠠⠠⠁⠇⠇ ⠠⠠⠁⠗⠕⠥⠝⠙ ⠠⠠⠍⠑
⠠⠠⠁⠇⠇ ⠠⠠⠁⠗⠕⠥⠝⠙ ⠠⠠⠍⠑ ⠠⠠⠁⠇⠇

⠠⠠
⠠⠠⠠

⠠⠊ ⠞⠓⠑ ⠙⠑⠉⠊⠎⠊⠕⠝ ⠺⠁⠎ ⠍⠁⠙⠑, ⠁⠎⠞⠕⠝⠊⠎⠓⠊⠝⠛ ⠋⠕⠗ ⠞⠓⠑
⠕⠝ ⠋⠊⠝⠁⠝⠉⠊⠁⠇ ⠍⠁⠗⠅⠑⠞⠎. ⠠⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠓⠁⠙ ⠃⠑⠑⠝
⠞⠗⠁⠙⠊⠝⠛ ⠃⠇⠥⠑ ⠉⠓⠊⠏⠎ ⠋⠕⠗ ⠞⠓⠗⠑⠑ ⠽⠑⠁⠗⠎, ⠁⠝⠙ ⠞⠓⠑⠽
⠕⠝⠇⠽ ⠓⠁⠙ ⠞⠓⠗⠑⠑ ⠉⠇⠊⠑⠝⠞⠎ ⠁⠞ ⠞⠓⠑ ⠞⠊⠍⠑ ⠕⠋ ⠞⠓⠑
⠊⠝⠉⠕⠍⠏⠁⠝⠽⠄ ⠠⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠁⠎ ⠁⠇⠎⠕ ⠞⠓⠑ ⠋⠊⠗⠎⠞
⠞⠕ ⠊⠝⠞⠗⠕⠙⠥⠉⠑ ⠁ ⠉⠕⠍⠏⠥⠞⠑⠗ ⠞⠕ ⠞⠓⠑ ⠍⠁⠗⠅⠑⠞⠄ ⠠⠞⠓⠑
⠉⠕⠍⠏⠁⠝⠽ ⠺⠁⠎ ⠋⠕⠥⠝⠙⠑⠙ ⠊⠝ ⠁⠏⠗⠊⠇ ⠁⠎ ⠁ ⠗⠑⠎⠥⠇⠞ ⠕⠋ ⠞⠓⠑
⠍⠑⠗⠛⠑⠗ ⠕⠋ ⠞⠓⠗⠑⠑ ⠉⠕⠍⠏⠁⠝⠊⠑⠎⠄ ⠠⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠁⠎
⠊⠝⠉⠕⠗⠏⠕⠗⠁⠞⠑⠙ ⠊⠝ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠙⠑⠇⠁⠺⠁⠗⠑⠄ ⠠⠞⠓⠑
⠉⠕⠍⠏⠁⠝⠽ ⠓⠁⠙ ⠁ ⠗⠑⠛⠊⠎⠞⠑⠗⠑⠙ ⠕⠋⠋⠊⠉⠑ ⠁⠞ ⠞⠓⠑ ⠉⠁⠏⠊⠞⠁⠇
⠕⠋ ⠙⠑⠇⠁⠺⠁⠗⠑⠄ ⠠⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠁⠎ ⠁⠇⠎⠕ ⠞⠓⠑ ⠋⠊⠗⠎⠞
⠞⠕ ⠊⠝⠞⠗⠕⠙⠥⠉⠑ ⠁ ⠉⠕⠍⠏⠥⠞⠑⠗ ⠞⠕ ⠞⠓⠑ ⠍⠁⠗⠅⠑⠞⠄

⠠⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠁⠎ ⠋⠕⠥⠝⠙⠑⠙ ⠃⠽ ⠠⠉⠓⠁⠗⠇⠑⠎ ⠠⠙⠁⠗⠺⠊⠝⠄ ⠠⠓⠑
⠺⠁⠎ ⠁ ⠝⠁⠞⠥⠗⠁⠇⠊⠎⠞ ⠁⠝⠙ ⠁ ⠉⠕⠍⠏⠁⠝⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠞⠊⠍⠑⠎⠄ ⠠⠓⠑
⠙⠑⠎⠉⠗⠊⠃⠑⠙ ⠞⠓⠑ ⠕⠗⠊⠛⠊⠝ ⠕⠋ ⠎⠏⠑⠉⠊⠑⠎ ⠁⠎ ⠁ ⠗⠑⠎⠥⠇⠞ ⠕⠋ ⠝⠁⠞⠥⠗⠁⠇
⠎⠑⠇⠑⠉⠞⠊⠕⠝⠄ ⠠⠓⠑ ⠁⠇⠎⠕ ⠙⠑⠎⠉⠗⠊⠃⠑⠙ ⠞⠓⠑ ⠕⠗⠊⠛⠊⠝ ⠕⠋ ⠞⠓⠑
⠙⠕⠍⠑⠎⠞⠊⠉ ⠉⠁⠞⠞⠇⠑⠄

The algorithm described in the previous section is the basis of many modern implementations. However, it has a number of limitations that we will address in this section. First, the algorithm is not very efficient for large inputs. Second, it does not handle duplicate values well. Third, it is not stable, meaning that the relative order of equal elements may change.

To address these issues, we can use a more sophisticated approach. One such approach is the merge algorithm, which we will describe in detail below. The merge algorithm has a time complexity of O(n log n), which is much better than the O(n^2) complexity of the simple algorithm.

3.1 Merging Two Sorted Lists

The key insight behind the merge algorithm is that two sorted lists can be combined into a single sorted list in linear time. To see how this works, consider the following example:

```
List A: [1, 3, 5, 7]
List B: [2, 4, 6, 8]
Result: [1, 2, 3, 4, 5, 6, 7, 8]
```

We start by comparing the first element of each list. The smaller of the two is the first element of the result. We then advance the pointer in the list that contained the smaller element and repeat the process. When one of the lists is exhausted, we append the remaining elements of the other list to the result.

Here is a Python implementation of the merge function:

```
def merge(a, b):
    result = []
    i = j = 0
    while i < len(a) and j < len(b):
        if a[i] <= b[j]:
            result.append(a[i])
            i += 1
        else:
            result.append(b[j])
            j += 1
    result.extend(a[i:])
    result.extend(b[j:])
    return result
```

The time complexity of this function is O(n + m), where n and m are the lengths of the two input lists. This is because we visit each element exactly once.

3.2 Merge Sort

Merge sort is a divide-and-conquer algorithm that uses the merge function as a subroutine. The idea is to split the input list in half, recursively sort each half, and then merge the two sorted halves. The base case is a list of zero or one elements, which is already sorted.

```
def merge_sort(lst):
    if len(lst) <= 1:
        return lst
    mid = len(lst) // 2
    left = merge_sort(lst[:mid])
    right = merge_sort(lst[mid:])
    return merge(left, right)
```

The time complexity of merge sort is O(n log n), which is optimal for comparison-based sorting algorithms. The space complexity is O(n), because we need to store the temporary lists during the merge step.

Merge sort is stable, meaning that the relative order of equal elements is preserved. This is important in many applications, for example when sorting records by multiple keys.

3.3 Quicksort

Quicksort is another divide-and-conquer algorithm that is widely used in practice. It works by selecting a pivot element, partitioning the list into elements less than the pivot and elements greater than the pivot, and then recursively sorting the two partitions.

```
def quicksort(lst):
    if len(lst) <= 1:
        return lst
    pivot = lst[len(lst) // 2]
    left = [x for x in lst if x < pivot]
    middle = [x for x in lst if x == pivot]
    right = [x for x in lst if x > pivot]
    return quicksort(left) + middle + quicksort(right)
```

The average time complexity of quicksort is O(n log n), but the worst case is O(n^2). The worst case occurs when the pivot is always the smallest or largest element, for example when the list is already sorted. To avoid this, we can choose the pivot randomly or use the median of three elements.

Quicksort is not stable, but it is often faster than merge sort in practice because it has better cache locality and requires less memory.

3.4 Heapsort

Heapsort is a comparison-based sorting algorithm that uses a binary heap data structure. It has a time complexity of O(n log n) and a space complexity of O(1), making it an in-place algorithm. However, it is not stable and is usually slower than quicksort in practice.

The heapsort algorithm first builds a max heap from the input list, and then repeatedly extracts the maximum element and places it at the end of the list. The heap is then rebuilt with the remaining elements.

```
def heapsort(lst):
    n = len(lst)
    for i in range(n // 2 - 1, -1, -1):
        heapify(lst, n, i)
    for i in range(n - 1, 0, -1):
        lst[0], lst[i] = lst[i], lst[0]
        heapify(lst, i, 0)
    return lst

def heapify(lst, n, i):
    largest = i
    left = 2 * i + 1
    right = 2 * i + 2
    if left < n and lst[left] > lst[largest]:
        largest = left
    if right < n and lst[right] > lst[largest]:
        largest = right
    if largest != i:
        lst[i], lst[largest] = lst[largest], lst[i]
        heapify(lst, n, largest)
```

3.5 Comparison of Sorting Algorithms

The following table summarizes the properties of the sorting algorithms discussed in this chapter:

Algorithm	Time (average)	Time (worst)	Space	Stable
Merge sort	O(n log n)	O(n log n)	O(n)	Yes
Quicksort	O(n log n)	O(n^2)	O(log n)	No
Heapsort	O(n log n)	O(n log n)	O(1)	No

In practice, quicksort is often the fastest algorithm for large inputs, but merge sort is preferred when stability is required. Heapsort is rarely used in practice because it is slower than both quicksort and merge sort.

⠀⠀⠀

: ⠠⠇⠊⠎⠞ ⠙ ⠏⠗⠊⠝⠞ ⠙⠊⠎⠞⠗⠊⠃⠥⠞⠊⠕⠝ ⠙ ⠉⠁⠇⠊⠃⠑⠗ ⠕⠋ ⠞⠓⠑ ⠏⠗⠑⠎⠙⠑⠝⠞⠊⠁⠇
⠎ ⠏⠗⠑⠎⠊⠙⠑⠝⠞⠊⠁⠇ ⠉⠁⠍⠏⠁⠊⠛⠝ ⠕⠋ ⠞⠓⠑ ⠞⠺⠑⠝⠞⠊⠑⠞⠓ ⠉⠑⠝⠞⠥⠗⠽ ⠁⠝⠙ ⠞⠓⠑ ⠎⠥⠉⠉⠑⠎⠎⠊⠕⠝
⠕⠋ ⠁ ⠎⠞⠁⠞⠑ ⠕⠋ ⠞⠓⠑ ⠥⠝⠊⠕⠝ ⠕⠋ ⠎⠞⠁⠞⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠙⠊⠎⠞⠗⠊⠉⠞ ⠕⠋ ⠉⠕⠇⠥⠍⠃⠊⠁
⠞⠓⠗⠕⠥⠛⠓ ⠞⠓⠑ ⠎⠑⠝⠁⠞⠑ ⠁⠝⠙ ⠓⠕⠥⠎⠑ ⠕⠋ ⠗⠑⠏⠗⠑⠎⠑⠝⠞⠁⠞⠊⠧⠑⠎ ⠕⠝ ⠞⠓⠑
⠙⠁⠞⠑ ⠕⠋ ⠞⠓⠊⠎ ⠁⠉⠞ ⠁⠝⠙ ⠞⠓⠑ ⠎⠑⠁⠇ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠍⠕⠝⠺⠑⠁⠇⠞⠓ ⠕⠋ ⠍⠁⠎⠎⠁⠉⠓⠥⠎⠑⠞⠞⠎
⠁⠗⠑ ⠓⠑⠗⠑⠃⠽ ⠙⠥⠇⠽ ⠁⠙⠧⠊⠎⠑⠙ ⠁⠝⠙ ⠊⠝⠋⠕⠗⠍⠑⠙ ⠞⠓⠁⠞ ⠁ ⠍⠑⠑⠞⠊⠝⠛ ⠕⠋ ⠞⠓⠑
⠎⠞⠁⠝⠙⠊⠝⠛ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠕⠝ ⠞⠓⠑ ⠏⠗⠕⠏⠕⠎⠁⠇ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠞⠕ ⠉⠓⠁⠝⠛⠑
⠞⠓⠑ ⠝⠁⠍⠑ ⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠞⠕ ⠁ ⠝⠑⠺ ⠝⠁⠍⠑ ⠁⠝⠙ ⠞⠓⠁⠞ ⠞⠓⠑ ⠝⠑⠺
⠝⠁⠍⠑ ⠎⠓⠁⠇⠇ ⠃⠑ ⠅⠝⠕⠺⠝ ⠁⠎ ⠞⠓⠑ ⠎⠞⠁⠝⠙⠁⠗⠙ ⠁⠞ ⠁⠇⠇ ⠞⠊⠍⠑⠎ ⠁⠝⠙ ⠁⠇⠇ ⠏⠇⠁⠉⠑⠎
⠞⠓⠁⠞ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠕⠗ ⠁⠝⠽ ⠕⠞⠓⠑⠗ ⠏⠑⠗⠎⠕⠝ ⠕⠗ ⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠕⠗ ⠁⠝⠽
⠕⠞⠓⠑⠗ ⠁⠛⠑⠝⠉⠽ ⠕⠗ ⠁⠥⠞⠓⠕⠗⠊⠞⠽ ⠁⠉⠞⠊⠝⠛ ⠕⠝ ⠃⠑⠓⠁⠇⠋ ⠕⠋ ⠁ ⠛⠕⠧⠑⠗⠝⠍⠑⠝⠞⠁⠇
⠁⠛⠑⠝⠉⠽ ⠕⠗ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠕⠋ ⠁ ⠎⠥⠃⠙⠊⠧⠊⠎⠊⠕⠝ ⠕⠋ ⠁ ⠛⠕⠧⠑⠗⠝⠍⠑⠝⠞⠁⠇

⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠊⠎ ⠞⠓⠑ ⠉⠕⠍⠏⠇⠑⠞⠑ ⠗⠑⠉⠕⠗⠙ ⠕⠋ ⠁⠇⠇ ⠞⠓⠑ ⠞⠗⠁⠝⠎⠁⠉⠞⠊⠕⠝⠎ ⠞⠓⠁⠞ ⠓⠁⠧⠑ ⠁⠗⠊⠎⠑⠝ ⠋⠗⠕⠍ ⠞⠓⠑ ⠑⠭⠑⠉⠥⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠊⠎ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠃⠽ ⠞⠓⠑ ⠁⠥⠞⠓⠕⠗⠊⠞⠽ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠝⠙ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠞⠓⠑ ⠎⠁⠍⠑ ⠁⠎ ⠞⠓⠕⠎⠑ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⁂

⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠊⠎ ⠞⠓⠑ ⠉⠕⠍⠏⠇⠑⠞⠑ ⠗⠑⠉⠕⠗⠙ ⠕⠋ ⠁⠇⠇ ⠞⠓⠑ ⠞⠗⠁⠝⠎⠁⠉⠞⠊⠕⠝⠎ ⠞⠓⠁⠞ ⠓⠁⠧⠑ ⠁⠗⠊⠎⠑⠝ ⠋⠗⠕⠍ ⠞⠓⠑ ⠑⠭⠑⠉⠥⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠊⠎ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠃⠽ ⠞⠓⠑ ⠁⠥⠞⠓⠕⠗⠊⠞⠽ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠕⠗ ⠕⠋ ⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞⠊⠕⠝ ⠠⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⁂

⠠⠠
⠠⠠⠠

⠀⠀⠀

⠠⠠⠠

⠠⠠
⠠⠠⠠

⠠⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠍⠁⠊⠝⠞⠁⠊⠝⠎ ⠁⠇⠇ ⠉⠕⠍⠏⠥⠞⠑⠗ ⠎⠽⠎⠞⠑⠍⠎ ⠁⠝⠙ ⠁⠇⠇ ⠙⠁⠞⠁ ⠞⠓⠁⠞ ⠁⠗⠑
⠎⠞⠕⠗⠑⠙ ⠕⠝ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽⠄⠎ ⠎⠑⠗⠧⠑⠗⠎⠄ ⠠⠊⠝ ⠁⠙⠙⠊⠞⠊⠕⠝ ⠞⠕ ⠞⠓⠑ ⠏⠗⠕⠞⠑⠉⠞⠊⠕⠝ ⠍⠑⠁⠎⠥⠗⠑⠎
⠍⠁⠝⠙⠁⠞⠑⠙ ⠃⠽ ⠞⠓⠑ ⠠⠎⠁⠃⠗⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠠⠁⠉⠉⠑⠎⠎ ⠠⠉⠕⠝⠞⠗⠕⠇ ⠠⠁⠉⠞ ⠠⠁⠇⠇ ⠁
⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠉⠕⠍⠏⠥⠞⠑⠗ ⠎⠽⠎⠞⠑⠍⠎ ⠁⠝⠙ ⠁⠇⠇ ⠙⠁⠞⠁ ⠞⠓⠁⠞ ⠁⠗⠑ ⠎⠞⠕⠗⠑⠙
⠠⠕⠝ ⠞⠓⠑ ⠠⠎⠁⠃⠗⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠠⠁⠉⠉⠑⠎⠎ ⠠⠉⠕⠝⠞⠗⠕⠇ ⠠⠁⠉⠞ ⠠⠁⠇⠇ ⠁⠉⠉⠑⠎⠎
⠁⠞ ⠁ ⠞⠊⠍⠑ ⠃⠽ ⠞⠓⠑ ⠠⠎⠁⠃⠗⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠠⠁⠉⠉⠑⠎⠎ ⠠⠉⠕⠝⠞⠗⠕⠇ ⠠⠁⠉⠞ ⠠⠁⠇⠇
⠁⠉⠉⠑⠎⠎ ⠁⠞ ⁂

⠠⠠⠠

⠠⠠⠠

⠠⠠
⠠⠠⠠

⠠⠁⠏⠞⠁ ⠓⠁⠃⠊⠞⠁ ⠠⠎⠊ ⠊⠝⠉⠥⠗⠗⠑ ⠉⠁⠥⠎⠁ ⠙⠊ ⠉⠕⠍⠏⠕⠗⠞⠁⠍⠑⠝⠞⠕ ⠊
⠠⠙⠁⠍⠕ ⠙⠊ ⠠⠙⠊⠗⠊⠞⠞⠕ ⠉⠁⠏⠕ ⠁. ⠠⠙⠁⠃⠁⠗ ⠠⠙⠁⠃⠁⠗ ⠠⠁⠍⠁⠗⠁⠍⠑⠝⠞⠑
⠁ ⠠⠎⠑ ⠠⠊⠝ ⠠⠙⠁⠍⠕ ⠠⠍⠁ ⠠⠊⠝⠙⠁⠍⠁ ⠙⠊ ⠉⠕⠍⠏⠁⠎⠎⠊⠕⠝⠑ ⠠⠆⠂⠂⠂⠄
⠠⠊ ⠠⠁⠍⠁. ⠠⠍⠁⠎⠞⠑⠗⠉⠁⠗⠙⠕ ⠙⠊ ⠠⠁⠗⠞⠑⠎⠁⠝⠁⠞⠕ ⠙⠊ ⠠⠁⠍⠁⠗⠁⠍⠑⠝⠞⠑ ⠠⠁
⠠⠁⠍⠁⠗⠁⠍⠑⠝⠞⠑ ⠠⠆⠂⠂⠂⠄ ⠠⠍⠁ ⠠⠊⠝⠙⠁⠍⠁ ⠠⠍⠁⠎⠞⠑⠗⠉⠁⠗⠙⠕ ⠠⠁⠗⠞⠑⠎⠁⠝⠁⠞⠕
⠠⠙⠊ ⠠⠁⠍⠁⠗⠁⠍⠑⠝⠞⠑ ⠠⠁⠍⠁⠗⠁⠍⠑⠝⠞⠑ ⠠⠁⠍⠁⠗⠁⠍⠑⠝⠞⠑ ⠠⠁⠍⠁⠗⠁⠍⠑⠝⠞⠑

⠠⠠⠠

⠠⠃⠠⠇ ⠠⠁ ⠠⠠⠁⠠ ⠠⠁ ⠠⠊⠠⠇⠃⠠⠁ ⠠⠇ ⠠⠠ ⠠⠠⠁⠠⠠⠁⠠⠁⠠⠠ ⠠⠠⠁ ⠠⠃
⠠⠁⠠⠃⠇⠠⠁⠠⠁⠠⠊ ⠠⠠⠁⠠⠁⠠⠠⠁⠠⠃⠠⠇ ⠠⠁ ⠃⠠⠁⠠⠁⠠⠃⠠⠠⠁ ⠠⠁⠠ ⠠⠁⠠⠊⠠
⠠⠁⠠⠁⠠⠇ ⠇ ⠠⠠⠁⠠⠠⠁⠠⠁⠠⠠ ⠠⠁⠠ ⠁⠠⠁⠠⠠⠁⠠⠃⠠⠇⠠ ⠠⠁ ⠠⠁⠠⠊⠠⠁⠠⠃⠠⠇⠠ ⠠⠁ ⠠⠁⠠⠊⠠

⠠⠊⠠⠁ ⠠⠁⠠⠇⠠⠁ ⠠⠁ ⠠⠊⠠⠁⠠⠃⠠⠁

⠠⠠
⠠⠠⠠

[illegible]

[illegible]

.

[illegible]

[illegible]

⠠⠞⠓⠑ ⠞⠗⠁⠙⠑ ⠉⠕⠍⠍⠊⠎⠎⠊⠕⠝ ⠊⠎ ⠁ ⠋⠑⠙⠑⠗⠁⠇ ⠁⠛⠑⠝⠉⠽ ⠞⠓⠁⠞
⠺⠁⠎ ⠉⠗⠑⠁⠞⠑⠙ ⠃⠽ ⠞⠓⠑ ⠉⠕⠝⠛⠗⠑⠎⠎ ⠕⠋ ⠞⠓⠑ ⠥⠝⠊⠞⠑⠙ ⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠁⠍⠑⠗⠊⠉⠁
⠁⠝⠙ ⠊⠎ ⠗⠑⠎⠏⠕⠝⠎⠊⠃⠇⠑ ⠋⠕⠗ ⠞⠓⠑ ⠉⠕⠇⠇⠑⠉⠞⠊⠕⠝ ⠕⠋ ⠁ ⠇⠁⠗⠛⠑ ⠁⠍⠕⠥⠝⠞
⠕⠋ ⠙⠁⠞⠁ ⠞⠓⠁⠞ ⠺⠁⠎ ⠞⠓⠑⠝ ⠁⠝⠁⠇⠽⠵⠑⠙ ⠁⠝⠙ ⠊⠝⠞⠑⠗⠏⠗⠑⠞⠑⠙ ⠃⠽
⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠞⠕ ⠙⠑⠞⠑⠗⠍⠊⠝⠑ ⠊⠋ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠁⠎ ⠁⠃⠊⠙⠊⠝⠛
⠞⠓⠑ ⠞⠑⠗⠍⠎ ⠕⠋ ⠞⠓⠑ ⠁⠛⠗⠑⠑⠍⠑⠝⠞ ⠕⠗ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠁⠎ ⠁⠇⠎⠕
⠞⠕ ⠙⠑⠞⠑⠗⠍⠊⠝⠑ ⠊⠋ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠁⠎ ⠁⠃⠇⠑ ⠞⠕ ⠏⠁⠽ ⠞⠓⠑ ⠙⠑⠃⠞⠎
⠕⠋ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠁⠎ ⠞⠓⠑⠽ ⠃⠑⠉⠁⠍⠑ ⠙⠥⠑ ⠁⠝⠙ ⠏⠁⠽⠁⠃⠇⠑ ⠁⠞ ⠞⠓⠑
⠞⠊⠍⠑ ⠕⠋ ⠞⠓⠑ ⠁⠛⠗⠑⠑⠍⠑⠝⠞ ⠕⠗ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠁⠎ ⠙⠊⠎⠎⠕⠇⠧⠑⠙ ⠲

⠠⠠
⠠⠠⠠⠠⠠
⠠⠠⠠⠠⠠

⠠⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠

Chapter Introduction to Programming

Programming is the process of creating instructions that tell a computer how to perform a task. These instructions are written in a programming language, which is a formal language that both humans and computers can understand. Learning to program is like learning to speak a new language, but instead of communicating with people, you are communicating with machines.

There are many programming languages, each with its own strengths and weaknesses. Some languages are better suited for web development, while others are better for scientific computing or game development. The choice of language depends on the specific problem you are trying to solve and the tools available to you.

One of the most important concepts in programming is the idea of a variable. A variable is a named location in memory that can store a value. You can think of it as a box with a label on it. You can put different values in the box at different times, and you can refer to the box by its label to get the value that is currently stored in it.

Another important concept is the idea of a function. A function is a reusable piece of code that performs a specific task. Functions can take inputs, called arguments, and can return outputs. They are a way to organize your code and avoid repeating yourself.

Control flow is the order in which the instructions in a program are executed. By default, instructions are executed one after the other, from top to bottom. However, you can use control flow statements to change this order, for example to execute a block of code only if a certain condition is true, or to repeat a block of code a certain number of times.

Data structures are ways of organizing and storing data in memory. Some common data structures are arrays, lists, dictionaries, and trees. The choice of data structure can have a significant impact on the performance of your program, so it is important to choose the right one for the job.

Debugging is the process of finding and fixing errors in your code. Errors can be syntax errors, which are violations of the rules of the programming language, or logic errors, which are mistakes in the way you have designed your program. Debugging can be frustrating, but it is an essential skill for any programmer.

Testing is the process of verifying that your code works as expected. You can write tests that automatically check the behavior of your code, and you can run these tests every time you make a change to your code to make sure that you have not introduced any new errors. Testing is a good way to gain confidence in your code and to prevent bugs from creeping in.

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. It is an essential tool for any programmer who works on a project with other people, as it allows you to collaborate on the same code without overwriting each other's changes. It is also useful for working on your own projects, as it allows you to experiment with new ideas without fear of breaking your code.

Documentation is the process of writing down how your code works. Good documentation is clear, concise, and up to date. It is an important part of any project, as it helps other people understand your code and it helps you remember how your code works when you come back to it after a long time. Documentation can be written in comments in the code itself, or in separate files.

Algorithms are step-by-step procedures for solving a problem. They are the heart of computer science, and they are used in every program that you write. Some algorithms are simple, while others are very complex. The study of algorithms is a large and important field of computer science.

Object-oriented programming is a programming paradigm that organizes code around objects, which are instances of classes. A class is a blueprint for creating objects, and it defines the properties and behaviors that objects of that class will have. Object-oriented programming is a powerful way to model real-world problems in code.

Functional programming is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids changing state and mutable data. It is a declarative programming paradigm, which means that you describe what you want to do, rather than how to do it. Functional programming can be very elegant and concise, and it is well suited for parallel and concurrent programming.

Concurrency is the ability of a program to do more than one thing at the same time. This can be achieved by using multiple threads of execution, or by using asynchronous programming techniques. Concurrency can make your program more responsive and more efficient, but it can also be very difficult to get right.

Parallelism is the ability of a program to run on multiple processors at the same time. This can be achieved by using multiple threads of execution, or by using specialized hardware such as graphics processing units. Parallelism can make your program much faster, but it can also be very difficult to get right.

Networking is the ability of a program to communicate with other programs over a network. This can be achieved by using sockets, or by using higher-level protocols such as HTTP. Networking is the foundation of the internet, and it is an essential skill for any programmer who wants to build web applications or distributed systems.

Security is the process of protecting your program and your data from unauthorized access. This can be achieved by using encryption, by using authentication and authorization, and by following secure coding practices. Security is a very important topic, and it is one that every programmer should be aware of.

Performance is the measure of how fast your program runs and how much memory it uses. Performance can be improved by using efficient algorithms and data structures, by avoiding unnecessary work, and by using profiling tools to find the bottlenecks in your code. Performance is a very important topic, and it is one that every programmer should be aware of.

Maintainability is the measure of how easy it is to change your program. Maintainability can be improved by writing clear and concise code, by writing good documentation, and by writing tests. Maintainability is a very important topic, and it is one that every programmer should be aware of.

Portability is the measure of how easy it is to run your program on different platforms. Portability can be improved by using platform-independent languages and libraries, and by avoiding platform-specific code. Portability is a very important topic, and it is one that every programmer should be aware of.

Usability is the measure of how easy it is for people to use your program. Usability can be improved by designing a clear and intuitive user interface, by providing good documentation, and by testing your program with real users. Usability is a very important topic, and it is one that every programmer should be aware of.

Accessibility is the measure of how easy it is for people with disabilities to use your program. Accessibility can be improved by following accessibility guidelines, by using assistive technologies, and by testing your program with people with disabilities. Accessibility is a very important topic, and it is one that every programmer should be aware of.

Internationalization is the process of designing your program so that it can be easily adapted to different languages and cultures. This can be achieved by using Unicode, by separating the text from the code, and by using localization libraries. Internationalization is a very important topic, and it is one that every programmer should be aware of.

Localization is the process of adapting your program to a specific language and culture. This can be achieved by translating the text, by changing the date and time formats, and by changing the currency symbols. Localization is a very important topic, and it is one that every programmer should be aware of.

⠀⠀⠀

⠠⠠
⠠⠠⠠⠠
⠠⠠⠠⠠

⠠⠇⠕⠗⠑⠍ ⠊⠏⠎⠥⠍ ⠙⠕⠇⠕⠗ ⠎⠊⠞ ⠁⠍⠑⠞ ⠉⠕⠝⠎⠑⠉⠞⠑⠞⠥⠗ ⠁⠙⠊⠏⠊⠎⠉⠊⠝⠛ ⠑⠇⠊⠞
⠎⠑⠙ ⠙⠕ ⠑⠊⠥⠎⠍⠕⠙ ⠞⠑⠍⠏⠕⠗ ⠊⠝⠉⠊⠙⠊⠙⠥⠝⠞ ⠥⠞ ⠇⠁⠃⠕⠗⠑ ⠑⠞ ⠙⠕⠇⠕⠗⠑
⠍⠁⠛⠝⠁ ⠁⠇⠊⠟⠥⠁ ⠥⠞ ⠑⠝⠊⠍ ⠁⠙ ⠍⠊⠝⠊⠍ ⠧⠑⠝⠊⠁⠍ ⠟⠥⠊⠎ ⠝⠕⠎⠞⠗⠥⠙
⠑⠭⠻⠉⠊⠞⠁⠞⠊⠕⠝ ⠥⠇⠇⠁⠍⠉⠕⠗⠏⠑⠗ ⠎⠥⠎⠏⠊⠏⠊⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠭ ⠑⠁
⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠑⠭ ⠑⠁ ⠙⠕⠇⠕⠗⠑
⠍⠁⠛⠝⠁ ⠁⠇⠊⠟⠥⠁ ⠥⠞ ⠑⠝⠊⠍ ⠁⠙ ⠍⠊⠝⠊⠍ ⠧⠑⠝⠊⠁⠍ ⠟⠥⠊⠎
⠝⠕⠎⠞⠗⠥⠙ ⠑⠭⠻⠉⠊⠞⠁⠞⠊⠕⠝ ⠥⠇⠇⠁⠍⠉⠕⠗⠏⠑⠗ ⠎⠥⠎⠏⠊⠏⠊⠞ ⠁⠝⠊⠍
⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠭
⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕
⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁
⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕
⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞
⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁
⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕
⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁
⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕
⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞
⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁
⠉⠕⠍⠍⠕⠙⠕ ⠉⠥⠇⠏⠁ ⠙⠕⠍⠊⠝⠛⠕ ⠋⠥⠛⠊⠁⠞ ⠁⠝⠊⠍ ⠊⠙ ⠑⠁ ⠉⠕⠍⠍⠕⠙⠕
⠉⠥⠇⠏⠁

⠠⠠⠠⠞⠓⠑ ⠉⠗⠁⠃ ⠃⠁⠝⠅ ⠺⠁⠎ ⠁ ⠏⠗⠕⠎⠏⠑⠗⠕⠥⠎ ⠝⠑⠺ ⠙⠑⠧⠑⠇⠕⠏⠍⠑⠝⠞ ⠕⠋ ⠁
⠎⠞⠁⠞⠑⠲ ⠠⠃⠥⠞ ⠁⠎ ⠞⠓⠑ ⠽⠑⠁⠗⠎ ⠏⠁⠎⠎⠑⠙ ⠁ ⠎⠑⠗⠊⠑⠎ ⠕⠋ ⠙⠊⠎⠁⠎⠞⠑⠗⠎ ⠙⠗⠁⠺
⠞⠓⠑ ⠞⠗⠁⠙⠊⠞⠊⠕⠝⠁⠇ ⠏⠁⠞⠞⠑⠗⠝⠎ ⠕⠋ ⠇⠊⠋⠑ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠎⠏⠑⠗⠕⠥⠎ ⠺⠑⠇⠇⠲
⠠⠞⠓⠑ ⠉⠕⠍⠍⠥⠝⠊⠞⠽ ⠺⠁⠎ ⠎⠍⠁⠇⠇⠲ ⠠⠑⠧⠑⠗⠽⠕⠝⠑ ⠅⠝⠑⠺ ⠁⠇⠇ ⠞⠓⠑
⠊⠝⠞⠑⠗⠁⠉⠞⠊⠕⠝⠎ ⠁⠝⠙ ⠞⠓⠑ ⠏⠁⠞⠞⠑⠗⠝⠎ ⠕⠋ ⠇⠊⠋⠑ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠎⠏⠑⠗⠕⠥⠎
⠺⠑⠇⠇⠲

---

[1] ⠠⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠋⠕⠗ ⠠⠁⠙⠧⠁⠝⠉⠑⠍⠑⠝⠞ ⠕⠋ ⠠⠎⠉⠊⠑⠝⠉⠑
⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠊⠎ ⠁ ⠝⠕⠝⠏⠗⠕⠋⠊⠞ ⠕⠗⠛⠁⠝⠊⠵⠁⠞⠊⠕⠝ ⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠞⠕ ⠁⠙⠧⠁⠝⠉⠊⠝⠛
⠁⠝ ⠁⠉⠉⠥⠗⠁⠞⠑ ⠅⠝⠕⠺⠇⠑⠙⠛⠑ ⠕⠋ ⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠏⠗⠁⠉⠞⠊⠉⠑⠎ ⠁⠝⠙ ⠏⠕⠇⠊⠉⠊⠑⠎
⠋⠕⠗ ⠊⠝⠉⠗⠑⠁⠎⠊⠝⠛ ⠞⠓⠑ ⠧⠁⠇⠥⠑ ⠕⠋ ⠁⠉⠉⠕⠥⠝⠞⠁⠃⠊⠇⠊⠞⠽ ⠁⠝⠙
⠏⠗⠕⠋⠑⠎⠎⠊⠕⠝⠁⠇⠊⠎⠍ ⠊⠝ ⠁⠇⠇ ⠎⠑⠉⠞⠕⠗⠎ ⠕⠋ ⠏⠥⠃⠇⠊⠉ ⠁⠙⠍⠊⠝⠊⠎⠞⠗⠁⠞⠊⠕⠝ ⠁⠝⠙
⠁⠉⠉⠕⠥⠝⠞⠁⠃⠊⠇⠊⠞⠽ ⠊⠝ ⠞⠓⠑ ⠥⠝⠊⠞⠑⠙ ⠎⠞⠁⠞⠑⠎ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠋⠑⠎⠎⠊⠕⠝ ⠕⠋
⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠋⠕⠗ ⠠⠁⠙⠧⠁⠝⠉⠑⠍⠑⠝⠞ ⠕⠋ ⠠⠎⠉⠊⠑⠝⠉⠑
⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠊⠎ ⠁ ⠝⠕⠝⠏⠗⠕⠋⠊⠞ ⠕⠗⠛⠁⠝⠊⠵⠁⠞⠊⠕⠝ ⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠞⠕
⠁⠙⠧⠁⠝⠉⠊⠝⠛ ⠁⠝⠙ ⠙⠊⠎⠎⠑⠍⠊⠝⠁⠞⠊⠝⠛ ⠅⠝⠕⠺⠇⠑⠙⠛⠑ ⠁⠃⠕⠥⠞ ⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞
⠏⠗⠁⠉⠞⠊⠉⠑⠎ ⠁⠝⠙ ⠏⠕⠇⠊⠉⠊⠑⠎ ⠋⠕⠗ ⠊⠝⠉⠗⠑⠁⠎⠊⠝⠛ ⠞⠓⠑ ⠧⠁⠇⠥⠑ ⠕⠋ ⠁⠉⠉⠕⠥⠝⠞⠁⠃⠊⠇⠊⠞⠽
⠁⠝⠙ ⠊⠝⠉⠗⠑⠁⠎⠊⠝⠛ ⠞⠓⠑ ⠏⠗⠕⠋⠑⠎⠎⠊⠕⠝ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝
⠋⠕⠗ ⠠⠁⠙⠧⠁⠝⠉⠑⠍⠑⠝⠞ ⠕⠋ ⠠⠎⠉⠊⠑⠝⠉⠑ ⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠃⠥⠎⠊⠝⠑⠎⠎ ⠁⠝⠙
⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠋⠕⠗ ⠠⠁⠙⠧⠁⠝⠉⠑⠍⠑⠝⠞ ⠕⠋
⠠⠎⠉⠊⠑⠝⠉⠑ ⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠃⠥⠎⠊⠝⠑⠎⠎ ⠁⠝⠙ ⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠠⠁⠍⠑⠗⠊⠉⠁⠝
⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠋⠕⠗ ⠠⠁⠙⠧⠁⠝⠉⠑⠍⠑⠝⠞ ⠕⠋ ⠠⠎⠉⠊⠑⠝⠉⠑ ⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞
⠃⠥⠎⠊⠝⠑⠎⠎ ⠁⠝⠙ ⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠋⠕⠗

⠠⠁ ⠠⠉⠇⠁⠎⠎⠊⠋⠊⠉⠁⠞⠊⠕⠝ ⠠⠁⠉⠞ ⠕⠋ ⠠⠁⠥⠛⠥⠎⠞ ⠠⠁ ⠠⠁⠉⠞ ⠕⠋ ⠠⠁⠥⠛⠥⠎⠞

⠠⠞⠓⠑ ⠠⠉⠁⠍⠃⠗⠊⠙⠛⠑ ⠠⠙⠊⠉⠞⠊⠕⠝⠁⠗⠽ ⠕⠋ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠎ ⠠⠁⠝⠙ ⠠⠎⠉⠊⠑⠝⠉⠑
⠠⠏⠗⠕⠧⠊⠙⠑⠙ ⠠⠍⠁⠝⠽ ⠠⠞⠕⠏⠊⠉⠎ ⠠⠁⠝⠙ ⠠⠎⠥⠃⠞⠕⠏⠊⠉⠎ ⠠⠋⠗⠕⠍ ⠠⠞⠓⠑
⠠⠞⠊⠍⠑ ⠠⠕⠋ ⠠⠃⠗⠊⠁⠝ ⠠⠞⠥⠗⠝⠑⠗⠎ ⠠⠁⠝⠙ ⠠⠓⠊⠎⠞⠕⠗⠊⠁⠝⠎ ⠠⠞⠕
⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠁⠝⠙ ⠠⠊⠞⠎ ⠠⠎⠥⠃⠎⠊⠙⠊⠁⠗⠽ ⠠⠊⠎ ⠠⠞⠓⠑ ⠠⠉⠁⠍⠃⠗⠊⠙⠛⠑
⠠⠙⠊⠉⠞⠊⠕⠝⠁⠗⠽ ⠠⠕⠋ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠎ ⠠⠁⠝⠙ ⠠⠎⠉⠊⠑⠝⠉⠑ ⠠⠊⠞ ⠠⠊⠎ ⠠⠝⠕⠞
⠠⠞⠓⠑ ⠠⠏⠥⠗⠏⠕⠎⠑ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠙⠊⠉⠞⠊⠕⠝⠁⠗⠽ ⠠⠞⠕ ⠠⠙⠑⠋⠊⠝⠑ ⠠⠁⠇⠇
⠠⠞⠓⠑ ⠠⠞⠑⠗⠍⠎ ⠠⠁⠝⠙ ⠠⠉⠕⠝⠉⠑⠏⠞⠎ ⠠⠊⠝ ⠠⠞⠓⠑ ⠠⠙⠊⠉⠞⠊⠕⠝⠁⠗⠽ ⠠⠁⠗⠑
⠠⠞⠁⠅⠑⠝ ⠠⠋⠗⠕⠍ ⠠⠞⠓⠑ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠠⠁⠝⠙ ⠠⠎⠉⠊⠑⠝⠉⠑ ⠠⠎⠞⠁⠝⠙⠁⠗⠙
⠠⠞⠓⠑⠗⠑ ⠠⠊⠎ ⠠⠝⠕ ⠠⠏⠗⠕⠧⠊⠎⠊⠕⠝ ⠠⠞⠕ ⠠⠉⠓⠁⠝⠛⠑ ⠠⠞⠓⠑ ⠠⠍⠑⠁⠝⠊⠝⠛ ⠠⠕⠋

⠠⠁⠇⠇⠑⠛⠑ ⠁ ⠍⠁⠝ ⠺⠓⠕ ⠓⠁⠙ ⠞⠺⠕ ⠎⠞⠕⠝⠑ ⠃⠕⠁⠞⠎.
⠠⠠⠺⠓⠑⠝ ⠓⠑ ⠋⠕⠥⠝⠙

⠠⠠⠞⠓⠑ ⠠⠠⠉⠕⠍⠏⠁⠝⠊⠕⠝ ⠊⠎ ⠁ ⠏⠇⠁⠉⠑ ⠞⠓⠁⠞
⠁ ⠏⠗⠕⠍⠊⠎⠑⠙ ⠊⠎⠇⠁⠝⠙, ⠠⠠⠃⠥⠞ ⠊⠞ ⠙⠕⠑⠎⠝⠞ ⠁⠇⠺⠁⠽⠎ ⠍⠁⠅⠑
⠁ ⠏⠗⠕⠍⠊⠎⠑ ⠊⠞ ⠊⠎ ⠞⠕ ⠃⠑ ⠅⠑⠏⠞ ⠁ ⠏⠗⠕⠍⠊⠎⠑

⠠⠠
⠠⠠⠠⠠
⠠⠠⠠⠠

A Comprehensive Overview of Modern Approaches

Introduction

This section provides an introduction to the topic and sets the stage for the discussion that follows. We will explore various aspects in detail.

Background

Previous work has established a foundation for understanding these concepts. Researchers have made significant progress in recent years.

Methodology

Our approach combines several techniques to achieve the desired results. We describe each step in the following subsections.

Results

The findings demonstrate clear improvements over baseline methods. Tables and figures summarize the key observations.

Conclusion

In summary, we have presented a comprehensive analysis. Future work will address remaining limitations.

⠠⠞⠓⠑ ⠠⠏⠗⠕⠉⠑⠎⠎ ⠠⠕⠋ ⠠⠉⠕⠍⠏⠁⠉⠞ ⠠⠠ ⠠⠠⠠⠉⠁⠝ ⠠⠃⠑ ⠠⠥⠎⠑⠙ ⠠⠞⠕
⠠⠠⠠⠙⠑⠎⠉⠗⠊⠃⠑ ⠠⠁ ⠠⠠⠠⠏⠗⠕⠉⠑⠎⠎ ⠠⠕⠋ ⠠⠠⠠⠙⠑⠎⠉⠗⠊⠃⠊⠝⠛ ⠠⠁ ⠠⠠⠠⠏⠗⠕⠉⠑⠎⠎
⠠⠠⠠⠞⠓⠁⠞ ⠠⠠⠠⠁⠉⠓⠊⠑⠧⠑⠎ ⠠⠠⠠⠁ ⠠⠠⠠⠙⠑⠎⠊⠗⠑⠙ ⠠⠠⠠⠕⠥⠞⠉⠕⠍⠑⠲

⠠⠠⠠⠁⠉⠉⠕⠗⠙⠊⠝⠛ ⠠⠞⠕ ⠠⠠⠠⠞⠓⠑ ⠠⠠⠠⠏⠗⠕⠉⠑⠎⠎ ⠠⠠⠠⠙⠑⠎⠉⠗⠊⠃⠑⠙ ⠠⠊⠝ ⠠⠠⠠⠎⠑⠉⠞⠊⠕⠝
⠠⠠⠠⠁ ⠠⠠⠠⠕⠋ ⠠⠠⠠⠞⠓⠊⠎ ⠠⠠⠠⠙⠕⠉⠥⠍⠑⠝⠞⠲ ⠠⠠⠠⠁⠝ ⠠⠠⠠⠁⠉⠉⠕⠗⠙⠊⠝⠛
⠠⠠⠠⠞⠕ ⠠⠠⠠⠞⠓⠊⠎ ⠠⠠⠠⠏⠗⠕⠉⠑⠎⠎ ⠠⠠⠠⠁ ⠠⠠⠠⠏⠁⠗⠞⠽ ⠠⠠⠠⠎⠓⠁⠇⠇ ⠠⠠⠠⠙⠕
⠠⠠⠠⠎⠕ ⠠⠠⠠⠊⠝ ⠠⠠⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠠⠠⠺⠊⠞⠓ ⠠⠠⠠⠞⠓⠑ ⠠⠠⠠⠏⠁⠗⠞⠽⠄⠎
⠠⠠⠠⠏⠗⠊⠝⠉⠊⠏⠁⠇ ⠠⠠⠠⠏⠇⠁⠉⠑ ⠠⠠⠠⠕⠋ ⠠⠠⠠⠃⠥⠎⠊⠝⠑⠎⠎⠲

⠠⠠⠠⠁⠉⠉⠕⠗⠙⠊⠝⠛ ⠠⠞⠕ ⠠⠠⠠⠞⠓⠑ ⠠⠠⠠⠏⠗⠕⠉⠑⠎⠎ ⠠⠠⠠⠙⠑⠎⠉⠗⠊⠃⠑⠙ ⠠⠊⠝
⠠⠠⠠⠎⠑⠉⠞⠊⠕⠝ ⠠⠠⠠⠁ ⠠⠠⠠⠕⠋ ⠠⠠⠠⠞⠓⠊⠎ ⠠⠠⠠⠙⠕⠉⠥⠍⠑⠝⠞ ⠠⠠⠠⠁⠝ ⠠⠠⠠⠁⠉⠉⠕⠗⠙⠊⠝⠛
⠠⠠⠠⠞⠕ ⠠⠠⠠⠞⠓⠊⠎ ⠠⠠⠠⠏⠗⠕⠉⠑⠎⠎ ⠠⠠⠠⠁ ⠠⠠⠠⠏⠁⠗⠞⠽ ⠠⠠⠠⠎⠓⠁⠇⠇ ⠠⠠⠠⠙⠕
⠠⠠⠠⠎⠕ ⠠⠠⠠⠊⠝ ⠠⠠⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠠⠠⠺⠊⠞⠓ ⠠⠠⠠⠞⠓⠑ ⠠⠠⠠⠏⠁⠗⠞⠽⠄⠎
⠠⠠⠠⠏⠗⠊⠝⠉⠊⠏⠁⠇ ⠠⠠⠠⠏⠇⠁⠉⠑ ⠠⠠⠠⠕⠋ ⠠⠠⠠⠃⠥⠎⠊⠝⠑⠎⠎⠲

⠠⠠⠠⠁⠉⠉⠕⠗⠙⠊⠝⠛ ⠠⠞⠕ ⠠⠠⠠⠞⠓⠑ ⠠⠠⠠⠏⠗⠕⠉⠑⠎⠎ ⠠⠠⠠⠙⠑⠎⠉⠗⠊⠃⠑⠙ ⠠⠊⠝
⠠⠠⠠⠎⠑⠉⠞⠊⠕⠝ ⠠⠠⠠⠁ ⠠⠠⠠⠕⠋ ⠠⠠⠠⠞⠓⠊⠎ ⠠⠠⠠⠙⠕⠉⠥⠍⠑⠝⠞ ⠠⠠⠠⠁⠝ ⠠⠠⠠⠁⠉⠉⠕⠗⠙⠊⠝⠛
⠠⠠⠠⠞⠕ ⠠⠠⠠⠞⠓⠊⠎ ⠠⠠⠠⠏⠗⠕⠉⠑⠎⠎ ⠠⠠⠠⠁ ⠠⠠⠠⠏⠁⠗⠞⠽ ⠠⠠⠠⠎⠓⠁⠇⠇ ⠠⠠⠠⠙⠕
⠠⠠⠠⠎⠕ ⠠⠠⠠⠊⠝ ⠠⠠⠠⠁⠉⠉⠕⠗⠙⠁⠝⠉⠑ ⠠⠠⠠⠺⠊⠞⠓ ⠠⠠⠠⠞⠓⠑ ⠠⠠⠠⠏⠁⠗⠞⠽⠄⠎
⠠⠠⠠⠏⠗⠊⠝⠉⠊⠏⠁⠇ ⠠⠠⠠⠏⠇⠁⠉⠑ ⠠⠠⠠⠕⠋ ⠠⠠⠠⠃⠥⠎⠊⠝⠑⠎⠎⠲

⠠⠠⠠⠁⠉⠉⠕⠗⠙⠊⠝⠛⠇⠽

⠠ ⠙⠗⠑⠁ ⠕⠋ ⠞⠕ ⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠙ ⠍⠁⠞⠑ ⠕⠋ ⠙⠕⠎⠑⠙
⠇⠁⠞⠑⠗ ⠞⠕ ⠙⠗⠽⠊⠝⠛ ⠍⠁⠞⠑ ⠙⠗⠽⠊⠝⠛ ⠍⠁⠞⠑ ⠃⠗⠑⠁⠙. ⠠⠙⠁⠙ ⠸⠁⠠⠙
⠠⠍⠁ ⠃⠥⠞ ⠙⠁⠙ ⠍⠁⠁⠁

⠠⠠
⠠⠠⠠⠠
⠠⠠⠠⠠

⠈⠓⠗ ⠈⠺ ⠠⠞ ⠠⠞⠕⠏⠗⠑⠎⠑⠝⠞ ⠈⠕ ⠠⠞ ⠓⠑⠗⠎ ⠠⠍⠊⠞ ⠏⠗⠕⠏⠑⠗ ⠈⠕
⠈⠞ ⠠⠏⠈⠗⠑⠈⠎⠊⠙⠑⠝⠞⠊⠁⠇ ⠠⠓⠑⠁⠇⠞⠓. ⠠⠁⠍⠑⠗. ⠠⠊⠎ ⠈⠝⠕⠺⠈⠓⠊
⠠⠞⠓ ⠇⠁⠞ ⠠⠞⠕⠏⠈⠕⠏ ⠠⠓⠕⠏⠑⠎. ⠈⠎⠊⠝⠈⠕ ⠠⠏⠗⠑⠎⠊⠙⠑⠝⠞⠊⠁⠇ ⠈⠏ ⠠⠃ ⠠⠞⠓
⠠⠓⠕⠞⠏⠕⠞ ⠠⠞⠓⠑ ⠠⠏⠁⠙ ⠠⠓⠁⠎⠓⠑⠙ ⠠⠓⠊⠎⠞⠕⠗⠊⠉⠁⠇⠇⠽ ⠠⠞⠕⠏⠊⠉⠎⠁⠇⠇⠽ ⠈⠊ ⠏
⠠⠃⠁ ⠈⠎⠁⠽⠎. ⠠⠺ ⠈⠏⠗⠕⠃⠁⠃⠇⠽ ⠙⠁⠽⠎ ⠏⠗⠕⠙⠥⠉⠑ ⠃ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠗⠎
⠃ ⠈⠞ ⠓⠁⠗⠙⠺⠁⠗⠑ ⠠⠞⠓ ⠇⠁⠞ ⠓⠊⠎⠞⠕⠗⠊⠉⠁⠇⠇⠽ ⠠⠑ ⠃ ⠏⠁ ⠠⠞⠓⠑⠽
⠠⠏ ⠊⠈⠎⠠⠺⠁⠞⠉⠓ ⠠⠓⠊⠎⠞ ⠓⠁⠙ ⠠⠁ ⠈⠙⠁⠽ ⠠⠁ ⠈⠝⠕⠞ ⠏⠁ ⠠⠞⠓⠑
⠓⠁⠙⠁ ⠠⠞⠓⠁⠞ ⠠⠉⠓⠁⠗⠁⠉⠞⠑⠗⠊⠎⠞⠊⠉ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠗⠎. ⠠⠓⠊⠎
⠠⠞⠓⠑ ⠠⠺⠁⠎ ⠠⠃ ⠃ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠙ ⠈⠎⠞⠁⠝⠙⠁⠗⠙

⠀⠀⠀

⠠⠠⠠

" ⠀⠀ ⠕ ⠞⠞⠓⠑⠗⠎ ⠺⠓ ⠺⠗⠊⠞⠑ ⠞⠓⠑⠍ ⠙⠕ ⠝⠕⠞ ⠁⠉⠉⠑⠏⠞ ⠓ ⠁⠝ ⠑⠍⠏⠞⠽
⠺⠁⠇⠇⠑⠞ ⠎⠽⠍⠃⠕⠇ ⠞⠓⠁⠞ ⠊ ⠃⠑⠇⠊⠑⠧⠑ ⠊ ⠺⠊⠇⠇ ⠝⠕⠞ ⠙⠕ ⠁ ⠎⠞⠁⠝⠙⠁⠗⠙ ⠎⠑⠞ ⠕⠋ ⠓⠁⠃⠊⠞⠎ ⠀⠕ ⠓
⠎⠞⠁⠝⠙⠎⠂ ⠞⠓⠁⠞ ⠊⠎ ⠕ ⠎⠞⠁⠝⠙⠁⠗⠙⠲ ⠓ ⠀ ⠏⠗⠑⠋⠑⠗⠎⠆⠇⠽ ⠞⠕ ⠓⠊⠍ ⠕ ⠇
⠋⠕ ⠏⠗⠑⠎⠑⠝⠞⠁⠞⠊⠕⠝ ⠺⠓ ⠇⠁⠞⠑⠗ ⠞⠓⠁⠝ ⠁⠎⠅ ⠎⠞⠁⠝⠙⠁⠗⠙⠎ ⠕⠋ ⠁⠎⠅
⠕⠗ ⠞⠕ ⠇⠊⠅⠑ ⠓⠊⠍ ⠕ ⠞⠓⠑⠕⠗⠊⠞⠊⠉⠁⠇⠇⠽ ⠀ ⠎⠁⠽ ⠁⠎⠅⠑⠙ ⠇⠊⠅⠑ ⠀
⠗⠁⠞⠓⠑⠗ ⠕ ⠉⠕⠍⠏⠁⠗⠊⠎⠕⠝⠎ ⠁⠗⠑ ⠁⠇⠇ ⠎⠥⠉⠓ ⠏⠇⠁⠉⠑⠎ ⠇⠊⠅⠑⠎ ⠏⠇⠁⠉⠑⠎ ⠞⠕
⠇⠊⠅⠑ ⠞⠓⠁⠞ ⠉⠕⠍⠏⠁⠗⠊⠎⠕⠝⠎ ⠀ ⠀ ⠕ ⠀⠀ ⠀⠀ ⠀⠀

⠠⠠⠁⠎ ⠁ ⠠⠠⠃⠗⠊⠙⠛⠑ ⠠⠠⠃⠁⠝⠅ ⠠⠠⠏⠗⠑⠎⠊⠙⠑⠝⠞ ⠠⠠⠊⠝⠙⠊⠁⠝⠁ ⠠⠠⠙⠑⠉⠇⠁⠗⠑⠙ ⠠⠠⠍⠁⠗⠞⠁⠇ ⠠⠠⠇⠁⠺
⠠⠠⠕⠝ ⠠⠠⠁⠏⠗⠊⠇ ⠼⠁⠊⠂ ⠼⠃⠚⠚⠚ ⠠⠠⠁⠝⠙ ⠠⠠⠊⠞⠎ ⠠⠠⠁⠍⠑⠝⠙⠍⠑⠝⠞⠎ ⠠⠠⠁⠗⠑ ⠠⠠⠊⠝ ⠠⠠⠋⠥⠇⠇ ⠠⠠⠋⠕⠗⠉⠑
⠠⠠⠁⠝⠙ ⠠⠠⠁⠗⠑ ⠠⠠⠁ ⠠⠠⠏⠁⠗⠞ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠉⠕⠝⠎⠞⠊⠞⠥⠞⠊⠕⠝ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠥⠝⠊⠞⠑⠙ ⠠⠠⠎⠞⠁⠞⠑⠎
⠠⠠⠁⠝⠙ ⠠⠠⠞⠕ ⠠⠠⠁⠍⠑⠝⠙ ⠠⠠⠞⠓⠑ ⠠⠠⠎⠁⠍⠑ ⠠⠠⠁⠎ ⠠⠠⠞⠓⠑ ⠠⠠⠋⠊⠋⠞⠑⠑⠝⠞⠓ ⠠⠠⠁⠍⠑⠝⠙⠍⠑⠝⠞⠎
⠠⠠⠁⠎ ⠠⠠⠁ ⠠⠠⠺⠓⠕⠇⠑ ⠠⠠⠁⠝⠙ ⠠⠠⠋⠕⠗ ⠠⠠⠁ ⠠⠠⠇⠊⠍⠊⠞⠑⠙ ⠠⠠⠞⠊⠍⠑ ⠠⠠⠕⠝⠇⠽ ⠠⠠⠞⠕
⠠⠠⠁⠉⠓⠊⠑⠧⠑ ⠠⠠⠞⠓⠑ ⠠⠠⠋⠕⠇⠇⠕⠺⠊⠝⠛ ⠠⠠⠏⠥⠗⠏⠕⠎⠑⠎⠲

⠠⠠⠎⠑⠉⠞⠊⠕⠝ ⠼⠁⠲ ⠠⠠⠁⠗⠞⠊⠉⠇⠑⠎ ⠠⠠⠁⠝⠙ ⠠⠠⠏⠁⠗⠞⠎ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠊⠎ ⠠⠠⠁⠉⠞ ⠠⠠⠎⠓⠁⠇⠇
⠠⠠⠃⠑ ⠠⠠⠊⠝⠞⠑⠗⠏⠗⠑⠞⠑⠙ ⠠⠠⠁⠎ ⠠⠠⠋⠕⠇⠇⠕⠺⠎⠲ ⠠⠠⠁ ⠠⠠⠏⠑⠗⠎⠕⠝ ⠠⠠⠺⠓⠕ ⠠⠠⠓⠁⠎ ⠠⠠⠃⠑⠑⠝
⠠⠠⠁⠉⠉⠥⠎⠑⠙ ⠠⠠⠕⠋ ⠠⠠⠁ ⠠⠠⠉⠗⠊⠍⠑ ⠠⠠⠕⠗ ⠠⠠⠕⠋⠋⠑⠝⠎⠑ ⠠⠠⠎⠓⠁⠇⠇ ⠠⠠⠃⠑ ⠠⠠⠙⠑⠑⠍⠑⠙
⠠⠠⠁ ⠠⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠠⠕⠗ ⠠⠠⠁ ⠠⠠⠏⠑⠗⠎⠕⠝ ⠠⠠⠺⠓⠕ ⠠⠠⠓⠁⠎ ⠠⠠⠃⠑⠑⠝ ⠠⠠⠉⠕⠝⠧⠊⠝⠉⠑⠙
⠠⠠⠕⠋ ⠠⠠⠁ ⠠⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠠⠕⠗ ⠠⠠⠁ ⠠⠠⠏⠑⠗⠎⠕⠝ ⠠⠠⠺⠓⠕ ⠠⠠⠓⠁⠎ ⠠⠠⠃⠑⠑⠝ ⠠⠠⠉⠕⠝⠧⠊⠝⠉⠑⠙
⠠⠠⠕⠋ ⠠⠠⠁ ⠠⠠⠉⠗⠊⠍⠑ ⠠⠠⠕⠗ ⠠⠠⠕⠋⠋⠑⠝⠎⠑ ⠠⠠⠎⠓⠁⠇⠇ ⠠⠠⠃⠑ ⠠⠠⠙⠑⠑⠍⠑⠙ ⠠⠠⠁

⠀ ⠠⠽⠀ ⠠⠠⠀ ⠊⠍⠠⠃ ⠠⠠⠽ ⠽⠊ ⠠⠊⠠⠍⠃⠍⠊⠽⠀ ⠽ ⠠⠃⠃⠽ ⠠
⠍⠠⠃⠍⠠ ⠽⠀ ⠠⠠⠎⠠ ⠽ ⠠⠠⠽⠀ ⠍⠠⠽⠀ ⠽⠠⠠⠽⠀ ⠃⠠⠍⠃⠠⠃⠍⠎⠀
⠀ ⠠⠃⠍⠊⠎⠠⠀ ⠃ ⠊⠍⠠⠎⠠⠍⠀ ⠽⠠ ⠃ ⠃⠍⠃ ⠽ ⠠⠠⠃⠎⠀ ⠠⠠⠃⠍
⠀ ⠎⠠⠀ ⠠⠠⠃⠍⠀ ⠠⠠⠍⠀ ⠠⠃⠍⠃⠠⠍⠀ ⠽ ⠠⠎⠠⠍⠀ ⠽ ⠃
⠎⠠⠀ ⠽⠀ ⠠⠃⠍⠀ ⠠⠍⠀ ⠠⠃⠍⠃⠠⠍⠍⠃⠍⠀ ⠠⠀ ⠠⠃⠍⠀ ⠠⠍⠀ ⠠⠠⠍⠀
⠠⠃ ⠀ ⠠⠠⠍⠃⠍⠠⠃ ⠽ ⠠⠃⠃⠍⠀ ⠠⠍⠃⠍⠠⠍⠠⠃ ⠀ ⠍⠠⠍⠀ ⠀ ⠠⠍⠀ ⠠⠍⠀
⠠⠠⠎⠀ ⠎⠠⠀ ⠠⠠⠍⠀ ⠀ ⠠⠠⠠⠃⠍⠀ ⠀ ⠠⠍⠀ ⠠⠠⠍⠀
⠀ ⠠⠍⠃⠍⠠⠍⠀

⠠⠠⠠⠠ ⠞⠓⠑ ⠠⠠⠠⠠ ⠕⠋ ⠍⠁⠝⠽ ⠠⠠⠠⠠ ⠞⠓⠑ ⠠⠠⠠⠠ ⠞⠕ ⠠⠠⠠⠠ ⠞⠓⠑ ⠠⠠⠠⠠ ⠞⠓⠑ ⠠⠠⠠⠠ ⠞⠓⠑

⠠ ⠳⠠⠒ ⠒ ⠨⠠⠓ ⠳⠠⠒ ⠠⠒ ⠠⠳ ⠳⠠⠒⠠⠒⠠⠒⠠⠒ ⠠⠒ ⠠⠒⠠⠒⠠⠒⠠⠒ ⠳ ⠒ ⠠⠒
⠠⠒ ⠒⠠⠒⠠⠒⠠⠒ ⠳⠠⠒ ⠠⠒⠠⠳⠠⠒ ⠳⠠⠒⠠⠒⠠⠒⠠⠒⠠⠒ ⠳ ⠳⠠⠒ ⠳⠠⠒ ⠠⠒⠠⠒
⠠⠒ ⠳⠠⠒ ⠠⠒⠠⠒⠠⠒ ⠳ ⠒⠠⠒ ⠳ ⠒ ⠨⠠⠒⠠⠒⠠⠒ ⠠⠒⠠⠒⠠⠒⠠⠒ ⠠⠒⠠⠒⠠⠒
⠠⠒⠠⠒⠠⠒ ⠳⠠⠒⠠⠒⠠⠒⠠⠒ ⠠⠒⠠⠳⠠⠒⠠⠒⠠⠒⠠⠒ ⠠⠒ ⠳⠠⠒⠠⠒⠠⠒ ⠳⠠⠒⠠⠒⠠⠒
⠠⠒⠠⠒⠠⠒ ⠳⠠⠒ ⠒⠠⠒ ⠒⠠⠒ ⠠⠒⠠⠒⠠⠒ ⠳⠠⠒ ⠠⠒⠠⠒⠠⠒⠠⠒⠠⠒ ⠠⠒⠠⠒⠠⠒⠠⠒ ⠒
⠒ ⠳⠠⠒⠠⠒⠠⠒ ⠠⠒⠠⠒⠠⠒⠠⠒ ⠒⠒⠒⠒ ⠒⠳⠠⠒ ⠠⠒⠠⠒⠠⠒ ⠒ ⠳⠠⠒⠠⠒⠠⠒⠠⠒⠠⠒
⠠⠒ ⠒⠠⠒⠠⠒⠠⠒ ⠠⠒⠠⠒ ⠠⠒⠠⠒⠠⠒ ⠠⠒ ⠠⠒⠠⠒⠠⠒⠠⠒ ⠳⠠⠒⠠⠒⠠⠒ ⠳ ⠳
⠠⠒⠠⠒⠠⠒ ⠒⠠⠒⠠⠒⠠⠒⠠⠒ ⠠⠒⠠⠒⠠⠒⠠⠒ ⠳⠠⠒⠠⠒⠠⠒ ⠳ ⠒⠒⠒ ⠒ ⠳⠠⠒ ⠒⠠⠒⠠⠒

⠠⠊⠇⠇⠁ ⠠⠉⠁⠗⠍⠑⠝ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠊⠥⠎ ⠠⠏⠗⠕⠏⠑⠗⠞⠊⠥⠎ ⠠⠁⠝⠙ ⠠⠁
⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠇⠇⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠊⠑⠎ ⠠⠕⠋ ⠠⠞⠓⠑⠊⠗
⠠⠕⠺⠝⠑⠗⠎⠓⠊⠏⠲ ⠠⠊⠞ ⠠⠊⠎ ⠠⠞⠓⠑ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠁ ⠠⠏⠻⠎⠕⠝
⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞
⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽
⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎
⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽
⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁
⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞
⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽
⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞
⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞
⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽
⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁
⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞
⠠⠁⠇⠎⠕ ⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞ ⠠⠊⠎ ⠠⠝⠕⠞ ⠠⠁ ⠠⠏⠗⠕⠏⠑⠗⠞⠽ ⠠⠞⠓⠁⠞
⠠⠊⠎ ⠠⠝⠕⠞

⠠⠓ ⠎⠞⠗⠊⠝⠛⠎ ⠞⠓⠁⠞ ⠙⠕⠝⠄⠞ ⠓⠁⠧⠑ ⠁ ⠎⠏⠁⠉⠑ ⠊⠝ ⠞⠓⠑⠊⠗ ⠝⠁⠍⠑⠎ ⠕⠗
⠙⠑⠎⠉⠗⠊⠃⠑ ⠁ ⠞⠁⠎⠅ ⠞⠓⠁⠞ ⠉⠕⠝⠞⠁⠊⠝⠎ ⠁ ⠎⠏⠁⠉⠑ ⠊⠝ ⠞⠓⠑⠊⠗ ⠝⠁⠍⠑⠎
⠕⠗ ⠙⠑⠎⠉⠗⠊⠃⠑ ⠞⠓⠑ ⠞⠁⠎⠅ ⠞⠓⠁⠞ ⠉⠕⠝⠞⠁⠊⠝⠎ ⠁ ⠎⠏⠁⠉⠑ ⠊⠝ ⠞⠓⠑⠊⠗
⠝⠁⠍⠑⠎ ⠕⠗ ⠙⠑⠎⠉⠗⠊⠃⠑ ⠞⠓⠑ ⠎⠏⠁⠉⠑ ⠊⠞⠎⠑⠇⠋ ⠠⠠⠠

⠠⠊⠝⠙⠊⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠇⠁⠺ ⠠⠊⠝⠙⠊⠁⠝ ⠇⠁⠺ ⠁⠝⠙ ⠠⠊⠝⠙⠊⠁⠝ ⠇⠁⠺ ⠎⠁⠽ ⠞⠓⠁⠞ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠞⠓⠁⠞ ⠊⠎ ⠝⠕⠞ ⠁ ⠏⠁⠗⠞⠝⠑⠗⠎⠓⠊⠏ ⠁⠝⠙ ⠊⠎ ⠝⠕⠞ ⠁ ⠎⠕⠇⠑ ⠏⠗⠕⠏⠗⠊⠑⠞⠕⠗⠎⠓⠊⠏ ⠁⠝⠙ ⠊⠎ ⠝⠕⠞ ⠁ ⠏⠁⠗⠞⠝⠑⠗⠎⠓⠊⠏ ⠁⠝⠙ ⠊⠎ ⠝⠕⠞ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠞⠓⠁⠞

⠠⠁⠎ ⠃⠥⠎⠊⠝⠑⠎⠎ ⠛⠗⠕⠺⠎ ⠕⠧⠑⠗ ⠞⠓⠑ ⠉⠕⠥⠗⠎⠑ ⠕⠋ ⠞⠺⠕ ⠙⠑⠉⠁⠙⠑⠎, ⠞⠓⠑ ⠏⠗⠕⠉⠥⠗⠑⠍⠑⠝⠞ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠓⠁⠙ ⠃⠑⠉⠕⠍⠑ ⠁ ⠍⠁⠚⠕⠗ ⠉⠕⠝⠞⠗⠁⠉⠞⠕⠗ ⠋⠕⠗ ⠞⠓⠑ ⠉⠕⠝⠎⠞⠗⠥⠉⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠝⠑⠺ ⠍⠁⠗⠎⠓ ⠇⠊⠝⠅ ⠞⠗⠁⠝⠎⠊⠞ ⠎⠽⠎⠞⠑⠍. ⠠⠞⠓⠑ ⠏⠗⠕⠉⠥⠗⠑⠍⠑⠝⠞ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠓⠁⠙ ⠃⠑⠑⠝ ⠞⠓⠑ ⠏⠗⠊⠍⠁⠗⠽ ⠕⠺⠝⠑⠗ ⠕⠋ ⠞⠓⠑ ⠗⠕⠁⠙ ⠞⠓⠗⠕⠥⠛⠓ ⠞⠓⠑ ⠏⠁⠎⠞ ⠞⠺⠕ ⠙⠑⠉⠁⠙⠑⠎, ⠃⠥⠞ ⠞⠓⠑ ⠗⠕⠁⠙ ⠺⠁⠎ ⠝⠑⠧⠑⠗ ⠙⠑⠎⠊⠛⠝⠑⠙ ⠋⠕⠗ ⠞⠓⠑ ⠍⠁⠎⠎⠊⠧⠑ ⠞⠗⠁⠋⠋⠊⠉ ⠞⠓⠁⠞ ⠊⠞ ⠝⠕⠺ ⠉⠁⠗⠗⠊⠑⠎. ⠠⠁⠎ ⠁ ⠗⠑⠎⠥⠇⠞, ⠞⠓⠑ ⠏⠗⠕⠉⠥⠗⠑⠍⠑⠝⠞ ⠊⠎ ⠉⠕⠝⠎⠞⠁⠝⠞⠇⠽ ⠃⠑⠊⠝⠛ ⠗⠑⠏⠁⠊⠗⠑⠙ ⠁⠝⠙ ⠞⠓⠑ ⠞⠗⠁⠋⠋⠊⠉ ⠊⠎ ⠞⠓⠥⠎ ⠞⠕ ⠙⠊⠧⠑⠗⠞⠑⠙ ⠁⠗⠕⠥⠝⠙ ⠞⠓⠑ ⠞⠺⠕ ⠙⠑⠉⠁⠙⠑⠎ ⠕⠋ ⠞⠓⠑ ⠏⠗⠕⠉⠥⠗⠑⠍⠑⠝⠞ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠕⠏⠑⠗⠁⠞⠊⠕⠝⠎ ⠞⠕⠙⠁⠽.

⠠⠗⠑⠉⠕⠍⠍⠑⠝⠙⠁⠞⠊⠕⠝⠎

⠠⠊⠝ ⠞⠓⠑ ⠇⠁⠎⠞ ⠞⠺⠕ ⠙⠑⠉⠁⠙⠑⠎, ⠺⠑ ⠓⠁⠧⠑ ⠎⠑⠑⠝ ⠞⠓⠑ ⠏⠗⠕⠉⠥⠗⠑⠍⠑⠝⠞ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠞⠁⠅⠑ ⠁ ⠎⠞⠁⠝⠙ ⠁⠛⠁⠊⠝⠎⠞ ⠞⠓⠑ ⠉⠕⠝⠎⠞⠗⠥⠉⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠝⠑⠺ ⠍⠁⠗⠎⠓ ⠇⠊⠝⠅ ⠞⠗⠁⠝⠎⠊⠞ ⠎⠽⠎⠞⠑⠍ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠏⠕⠎⠁⠇ ⠞⠕ ⠉⠕⠝⠧⠑⠗⠞ ⠞⠓⠑ ⠞⠗⠁⠋⠋⠊⠉ ⠞⠕ ⠁ ⠎⠊⠝⠛⠇⠑ ⠙⠊⠗⠑⠉⠞⠊⠕⠝ ⠕⠝ ⠞⠓⠑ ⠏⠗⠕⠉⠥⠗⠑⠍⠑⠝⠞ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠕⠏⠑⠗⠁⠞⠊⠕⠝⠎ ⠞⠕⠙⠁⠽.

⠠⠊⠝ ⠞⠓⠑ ⠎⠑⠉⠕⠝⠙ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠞⠁⠅⠑ ⠁ ⠎⠞⠁⠝⠙ ⠁⠛⠁⠊⠝⠎⠞ ⠞⠓⠑ ⠉⠕⠝⠎⠞⠗⠥⠉⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠝⠑⠺ ⠍⠁⠗⠎⠓ ⠇⠊⠝⠅ ⠞⠗⠁⠝⠎⠊⠞ ⠎⠽⠎⠞⠑⠍ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠏⠕⠎⠁⠇ ⠞⠕ ⠉⠕⠝⠧⠑⠗⠞ ⠞⠓⠑ ⠞⠗⠁⠋⠋⠊⠉ ⠞⠕ ⠁ ⠎⠊⠝⠛⠇⠑ ⠙⠊⠗⠑⠉⠞⠊⠕⠝ ⠕⠝ ⠞⠓⠑ ⠏⠗⠕⠉⠥⠗⠑⠍⠑⠝⠞ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠕⠏⠑⠗⠁⠞⠊⠕⠝⠎ ⠞⠕⠙⠁⠽.

⠠⠊⠝ ⠞⠓⠑ ⠞⠓⠊⠗⠙ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠞⠁⠅⠑ ⠁ ⠎⠞⠁⠝⠙ ⠁⠛⠁⠊⠝⠎⠞ ⠞⠓⠑ ⠉⠕⠝⠎⠞⠗⠥⠉⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠝⠑⠺ ⠍⠁⠗⠎⠓ ⠇⠊⠝⠅ ⠞⠗⠁⠝⠎⠊⠞ ⠎⠽⠎⠞⠑⠍ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠏⠕⠎⠁⠇ ⠞⠕ ⠉⠕⠝⠧⠑⠗⠞ ⠞⠓⠑ ⠞⠗⠁⠋⠋⠊⠉ ⠞⠕ ⠁ ⠎⠊⠝⠛⠇⠑ ⠙⠊⠗⠑⠉⠞⠊⠕⠝ ⠕⠝ ⠞⠓⠑ ⠏⠗⠕⠉⠥⠗⠑⠍⠑⠝⠞ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠕⠏⠑⠗⠁⠞⠊⠕⠝⠎ ⠞⠕⠙⠁⠽.

⠠⠁⠇⠇⠑⠛⠁⠞⠊⠕⠝ ⠞⠕ ⠞⠓⠑ ⠙⠑⠉⠊⠎⠊⠕⠝ ⠞⠕ ⠙⠑⠝⠽ ⠞⠓⠑ ⠍⠕⠞⠊⠕⠝ ⠋⠕⠗
⠞⠗⠁⠝⠎⠋⠑⠗ ⠕⠋ ⠞⠓⠑ ⠏⠗⠕⠏⠑⠗⠞⠽ ⠊⠝ ⠞⠓⠑ ⠉⠁⠎⠑ ⠕⠋ ⠁ ⠍⠕⠗⠞⠛⠁⠛⠑ ⠋⠕⠗⠑⠉⠇⠕⠎⠥⠗⠑
⠁⠝⠙ ⠎⠁⠇⠑ ⠕⠋ ⠞⠓⠑ ⠏⠗⠕⠏⠑⠗⠞⠽ ⠞⠕ ⠁ ⠞⠓⠊⠗⠙ ⠏⠁⠗⠞⠽

⠠⠺⠓⠑⠗⠑⠁⠎, ⠁⠇⠇ ⠏⠁⠗⠞⠊⠑⠎ ⠓⠁⠧⠑ ⠙⠑⠉⠇⠁⠗⠑⠙ ⠞⠓⠑⠊⠗ ⠊⠝⠞⠑⠝⠞⠊⠕⠝ ⠞⠕ ⠃⠑
⠃⠕⠥⠝⠙ ⠃⠽ ⠞⠓⠑ ⠞⠑⠗⠍⠎ ⠕⠋ ⠞⠓⠊⠎ ⠁⠛⠗⠑⠑⠍⠑⠝⠞,
⠁⠝⠙ ⠞⠓⠑ ⠉⠕⠥⠗⠞ ⠓⠁⠧⠊⠝⠛ ⠚⠥⠗⠊⠎⠙⠊⠉⠞⠊⠕⠝
⠕⠧⠑⠗ ⠞⠓⠑ ⠍⠁⠞⠞⠑⠗, ⠊⠎ ⠁⠙⠧⠊⠎⠑⠙ ⠞⠕
⠁⠏⠏⠑⠁⠇ ⠞⠕ ⠞⠓⠑ ⠎⠥⠏⠗⠑⠍⠑ ⠉⠕⠥⠗⠞.

⠠⠓⠑ ⠺⠁⠎ ⠁ ⠛⠗⠑⠁⠞ ⠍⠁⠝ ⠁⠝⠙ ⠓⠊⠎ ⠓⠁⠝⠙⠎ ⠺⠑⠗⠑ ⠎⠕ ⠎⠞⠗⠕⠝⠛ ⠞⠓⠁⠞ ⠓⠑ ⠉⠕⠥⠇⠙ ⠝⠕⠞ ⠃⠑ ⠃⠗⠕⠅⠑⠝ ⠃⠽ ⠁⠝⠽ ⠞⠓⠊⠝⠛ ⠁⠝⠙ ⠓⠑ ⠺⠁⠎ ⠝⠕⠞ ⠁ ⠍⠁⠝ ⠺⠓⠕ ⠓⠁⠙ ⠁ ⠎⠍⠁⠇⠇ ⠎⠞⠁⠞⠥⠑ ⠕⠋ ⠓⠊⠍⠎⠑⠇⠋ ⠁⠞ ⠁⠇⠇ ⠞⠊⠍⠑⠎ ⠁⠝⠙ ⠊⠝ ⠁⠇⠇ ⠏⠇⠁⠉⠑⠎ ⠠⠓⠑ ⠺⠁⠎ ⠁ ⠍⠁⠝ ⠺⠓⠕ ⠓⠁⠙ ⠁ ⠎⠍⠁⠇⠇ ⠎⠞⠁⠞⠥⠑ ⠕⠋ ⠓⠊⠍⠎⠑⠇⠋ ⠁⠞ ⠁⠇⠇ ⠞⠊⠍⠑⠎ ⠁⠝⠙ ⠊⠝ ⠁⠇⠇ ⠏⠇⠁⠉⠑⠎

⠠⠊⠝⠋⠇⠥⠢⠉⠊⠁ ⠠⠉⠕⠧⠊⠙ ⠠⠞⠑⠎⠞ ⠕⠋
⠠⠍⠁⠎⠞⠑⠗ ⠕⠋ ⠠⠍⠁⠚⠕⠗ ⠙⠊
⠠⠉⠗⠊⠞⠊⠉⠁⠇ ⠠⠁ ⠠⠃⠊⠕⠍⠑⠙⠊⠉⠁⠇
⠠⠍⠁⠗⠉⠓ ⠼⠃⠼⠚⠼⠁

⠠⠏⠗⠊⠽⠁ ⠠⠎⠓⠁⠗⠍⠁ ⠠⠠⠗⠁⠚⠁⠞ ⠠⠠⠙⠁⠗⠁

⠠⠞⠓⠊⠎ ⠏⠁⠏⠑⠗, ⠋⠊⠗⠎⠞ ⠁⠉⠉⠑⠏⠞⠑⠙ ⠁⠞ ⠠⠍⠊⠙⠙⠇⠑ ⠠⠑⠁⠎⠞ ⠠⠍⠑⠙⠊⠉⠁⠇
⠠⠉⠕⠇⠇⠑⠛⠑, ⠊⠎ ⠁ ⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑ ⠗⠑⠧⠊⠑⠺ ⠕⠋ ⠠⠍⠑⠙⠊⠉⠁⠇ ⠠⠑⠙⠥⠉⠁⠞⠊⠕⠝
⠊⠝ ⠊⠞⠎ ⠝⠁⠞⠊⠧⠑ ⠠⠊⠝⠙⠊⠁, ⠁⠝⠙ ⠊⠞ ⠑⠭⠏⠇⠁⠊⠝⠎ ⠁ ⠙⠊⠧⠑⠗⠎⠑ ⠗⠁⠝⠛⠑ ⠕⠋ ⠞⠕⠏⠊⠉⠎
⠁⠝⠙ ⠊⠎ ⠁ ⠏⠗⠕⠙⠥⠉⠞ ⠕⠋ ⠁ ⠇⠕⠝⠛ ⠁⠝⠙ ⠗⠊⠉⠓ ⠞⠗⠁⠙⠊⠞⠊⠕⠝ ⠕⠋ ⠉⠕⠍⠏⠁⠗⠁⠞⠊⠧⠑
⠠⠍⠑⠙⠊⠉⠁⠇ ⠗⠑⠎⠑⠁⠗⠉⠓ ⠊⠝ ⠞⠓⠑ ⠠⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎⠂ ⠠⠊⠞ ⠊⠎ ⠁⠇⠎⠕ ⠁ ⠉⠕⠍⠏⠁⠗⠁⠞⠊⠧⠑
⠗⠑⠧⠊⠑⠺ ⠕⠋ ⠞⠓⠑ ⠠⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠠⠍⠑⠙⠊⠉⠁⠇ ⠠⠎⠽⠎⠞⠑⠍⠂ ⠠⠞⠓⠑ ⠠⠠⠥⠝⠊⠞⠑⠙
⠠⠎⠞⠁⠞⠑⠎ ⠠⠍⠑⠙⠊⠉⠁⠇ ⠠⠎⠽⠎⠞⠑⠍ ⠊⠎ ⠁ ⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑ ⠎⠽⠎⠞⠑⠍ ⠞⠓⠁⠞
⠏⠗⠕⠧⠊⠙⠑⠎ ⠁⠉⠉⠑⠎⠎ ⠞⠕ ⠁ ⠗⠁⠝⠛⠑ ⠕⠋ ⠙⠊⠁⠛⠝⠕⠎⠞⠊⠉ ⠁⠝⠙ ⠞⠗⠑⠁⠞⠍⠑⠝⠞
⠎⠑⠗⠧⠊⠉⠑⠎ ⠁⠝⠙ ⠊⠎ ⠁⠉⠉⠑⠎⠎⠊⠃⠇⠑ ⠞⠕ ⠁⠇⠇ ⠁⠥⠞⠓⠕⠗⠊⠵⠑⠙ ⠥⠎⠑⠗⠎⠂
⠠⠞⠓⠑ ⠠⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠠⠍⠑⠙⠊⠉⠁⠇ ⠠⠎⠽⠎⠞⠑⠍ ⠊⠎ ⠁ ⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑
⠎⠽⠎⠞⠑⠍ ⠞⠓⠁⠞ ⠏⠗⠕⠧⠊⠙⠑⠎ ⠁⠉⠉⠑⠎⠎ ⠞⠕ ⠁ ⠗⠁⠝⠛⠑ ⠕⠋ ⠙⠊⠁⠛⠝⠕⠎⠞⠊⠉ ⠁⠝⠙
⠞⠗⠑⠁⠞⠍⠑⠝⠞ ⠎⠑⠗⠧⠊⠉⠑⠎ ⠁⠝⠙ ⠊⠎ ⠁⠉⠉⠑⠎⠎⠊⠃⠇⠑ ⠞⠕ ⠁⠇⠇ ⠁⠥⠞⠓⠕⠗⠊⠵⠑⠙
⠥⠎⠑⠗⠎⠲

⠠⠉⠓⠁⠏⠞⠑⠗ ⠼⠁

⠠⠞⠓⠊⠎ ⠊⠎ ⠁ ⠏⠗⠕⠧⠊⠎⠊⠕⠝⠁⠇ ⠙⠕⠉⠥⠍⠑⠝⠞⠂ ⠝⠕⠞ ⠋⠊⠝⠁⠇
⠁⠞ ⠁⠇⠇ ⠏⠕⠊⠝⠞⠎ ⠁⠝⠙ ⠊⠞ ⠊⠎ ⠎⠥⠃⠚⠑⠉⠞ ⠞⠕ ⠉⠓⠁⠝⠛⠑⠲
⠠⠊⠋ ⠠⠞⠓⠑⠗⠑ ⠁⠗⠑ ⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠊⠙⠑⠁⠎ ⠁⠃⠕⠥⠞ ⠞⠓⠊⠎ ⠞⠕⠏⠊⠉⠂
⠁⠝⠙ ⠁ ⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠝⠑⠑⠙⠑⠙⠂ ⠁ ⠝⠑⠺
⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞
⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞
⠁ ⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠓⠊⠗⠙
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠋⠕⠥⠗⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁
⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠋⠊⠋⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠊⠭⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠑⠧⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁
⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠊⠛⠓⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠝⠊⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁
⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠇⠑⠧⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠺⠑⠇⠋⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠓⠊⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂
⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁
⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠋⠕⠥⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺
⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠋⠊⠋⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞
⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞
⠁ ⠎⠊⠭⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠑⠧⠑⠝⠞⠑⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠊⠛⠓⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂
⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁
⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠝⠊⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺
⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠇⠑⠧⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠺⠑⠇⠋⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁
⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠓⠊⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞
⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞
⠁ ⠋⠕⠥⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠊⠭⠞⠑⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠑⠧⠑⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂
⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁
⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠊⠛⠓⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺
⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠝⠊⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞
⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞
⠁ ⠞⠓⠊⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠑⠧⠑⠝⠞⠑⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠊⠛⠓⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂
⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁
⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠝⠊⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺
⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠇⠑⠧⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠺⠑⠇⠋⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁
⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠓⠊⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞
⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞
⠁ ⠋⠕⠥⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠊⠭⠞⠑⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠑⠧⠑⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂
⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁
⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠊⠛⠓⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺
⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠝⠊⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞
⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞
⠁ ⠞⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑
⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠇⠑⠧⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠺⠑⠇⠋⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁
⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠓⠊⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞
⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞
⠁ ⠋⠕⠥⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠊⠭⠞⠑⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠑⠧⠑⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂
⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁
⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠊⠛⠓⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺
⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠝⠊⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞
⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞
⠁ ⠞⠓⠊⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠑⠧⠑⠝⠞⠑⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠊⠛⠓⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂
⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁
⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠝⠊⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺
⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠑⠇⠑⠧⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠺⠑⠇⠋⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁
⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞
⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠞⠓⠊⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞
⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞
⠁ ⠋⠕⠥⠗⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙
⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲ ⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠊⠭⠞⠑⠑⠝⠞⠓
⠞⠕⠏⠊⠉⠂ ⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑ ⠉⠗⠑⠁⠞⠑⠙⠲
⠠⠊⠋ ⠁ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠊⠎ ⠁⠃⠕⠥⠞ ⠁ ⠎⠑⠧⠑⠝⠞⠑⠑⠝⠞⠓ ⠞⠕⠏⠊⠉⠂
⠁ ⠝⠑⠺ ⠙⠕⠉⠥⠍⠑⠝⠞ ⠎⠓⠕⠥⠇⠙ ⠃⠑

⠠⠞⠓⠕⠍⠁⠎ ⠠⠚⠑⠋⠋⠑⠗⠎⠕⠝
⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠊⠕⠝ ⠕⠋ ⠠⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠏⠑⠕⠏⠇⠑
⠠⠁⠞ ⠠⠺⠁⠗

· ⠠⠠
⠠⠠⠠

⠠⠓⠊⠎⠞⠕⠗⠽ ⠕⠋ ⠠⠁⠗⠞ ⠊⠝ ⠠⠞⠓⠑ ⠠⠊⠝⠙⠥⠎⠞⠗⠊⠁⠇ ⠠⠁⠛⠑ ⠍⠁⠉⠓⠊⠝⠑⠎

⠠⠞⠓⠑ ⠍⠁⠉⠓⠊⠝⠑ ⠙⠑⠎⠊⠛⠝ ⠍⠁⠙⠑ ⠋⠗⠕⠍ ⠊⠞⠎ ⠋⠊⠗⠎⠞
⠕⠉⠉⠥⠏⠁⠞⠊⠕⠝ ⠞⠕ ⠏⠗⠕⠙⠥⠉⠑ ⠞⠓⠑ ⠏⠗⠕⠙⠥⠉⠞⠎ ⠞⠓⠁⠞ ⠺⠑⠗⠑ ⠝⠑⠑⠙⠑⠙ ⠁⠞
⠞⠓⠑ ⠞⠊⠍⠑ ⠕⠋ ⠞⠓⠑ ⠊⠝⠧⠑⠝⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠎⠞⠑⠁⠍ ⠢⠁⠔ ⠠⠁
⠉⠕⠍⠏⠇⠑⠞⠑⠇⠽ ⠝⠑⠺ ⠍⠁⠉⠓⠊⠝⠑ ⠺⠁⠎ ⠉⠕⠝⠎⠞⠗⠥⠉⠞⠑⠙ ⠋⠗⠕⠍
⠠⠁ ⠠⠁

⠠⠁ ⠠⠞⠓⠑⠕⠗⠽ ⠕⠋ ⠠⠁⠗⠞ ⠊⠝ ⠠⠞⠓⠑ ⠠⠊⠝⠙⠥⠎⠞⠗⠊⠁⠇ ⠠⠁⠛⠑ ⠍⠁⠉⠓⠊⠝⠑⠎ ⠠⠁
⠠⠓⠁⠗⠗⠊⠎ ⠠⠞⠓⠕⠍⠏⠎⠕⠝ ⠠⠁⠝⠙ ⠠⠊⠝ ⠠⠞⠓⠑ ⠠⠁⠛⠑ ⠕⠋ ⠠⠍⠁⠉⠓⠊⠝⠑⠎ ⠠⠞⠓⠑
⠠⠁⠗⠞ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠁⠗⠞⠊⠎⠞ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠁⠗⠞⠊⠎⠁⠝ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠁⠗⠞⠊⠎⠞
⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠁⠗⠞⠊⠎⠁⠝ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠁⠗⠞⠊⠎⠞ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠁⠗⠞⠊⠎⠁⠝

⠠⠓⠑ ⠠⠎⠁⠗⠁⠎ ⠕⠋ ⠁ ⠠⠉⠕⠍⠏⠁⠎⠎⠊⠕⠝ ⠓ ⠠⠞⠕⠎⠞⠁⠝⠞⠊⠝⠑⠠⠞⠎
⠠⠁⠗⠉⠓⠑⠗ ⠠⠁⠝⠙ ⠠⠍⠁⠝⠽ ⠞⠗⠊⠃⠥⠞⠁⠗⠽ ⠠⠍⠁⠛⠊⠉ ⠠⠊⠞⠑⠍⠎ ⠠⠁⠝⠙
⠠⠞⠓⠑⠊⠗ ⠠⠏⠕⠺⠑⠗⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠏⠕⠺⠑⠗⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑⠊⠗ ⠠⠏⠕⠺⠑⠗

⠠⠞⠗ ⠓⠁⠙ ⠞⠓⠗⠑⠑ ⠞⠊⠍⠑⠎ ⠞⠓⠑ ⠎⠞⠁⠋⠋ ⠞⠁⠇⠅⠑⠙ ⠁⠃⠕⠥⠞ ⠓⠕⠺ ⠞⠕ ⠍⠁⠅⠑
⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠥⠎⠑ ⠕⠋ ⠓⠊⠎ ⠞⠊⠍⠑ ⠊⠝ ⠁ ⠎⠞⠕⠗⠑ ⠁⠝⠙ ⠎⠕⠍⠑
⠞⠓⠊⠝⠛⠎ ⠞⠓⠁⠞ ⠺⠑⠗⠑ ⠝⠕⠞ ⠎⠕ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠺⠑⠗⠑ ⠞⠓⠑ ⠏⠑⠕⠏⠇⠑ ⠺⠓⠕ ⠺⠑⠗⠑
⠞⠓⠑ ⠎⠞⠁⠋⠋ ⠞⠁⠇⠅⠑⠙ ⠁⠃⠕⠥⠞ ⠓⠕⠺ ⠞⠕ ⠍⠁⠅⠑ ⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠥⠎⠑ ⠕⠋
⠓⠊⠎ ⠞⠊⠍⠑ ⠊⠝ ⠁ ⠎⠞⠕⠗⠑ ⠁⠝⠙ ⠎⠕⠍⠑⠞⠓⠊⠝⠛⠎ ⠞⠓⠁⠞ ⠺⠑⠗⠑ ⠝⠕⠞ ⠎⠕ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑
⠺⠑⠗⠑ ⠞⠓⠑ ⠏⠑⠕⠏⠇⠑ ⠺⠓⠕ ⠺⠑⠗⠑ ⠞⠓⠑ ⠎⠞⠁⠋⠋ ⠞⠁⠇⠅⠑⠙ ⠁⠃⠕⠥⠞ ⠓⠕⠺ ⠞⠕ ⠍⠁⠅⠑
⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠥⠎⠑ ⠕⠋ ⠓⠊⠎ ⠞⠊⠍⠑ ⠊⠝ ⠁ ⠎⠞⠕⠗⠑ ⠁⠝⠙ ⠎⠕⠍⠑⠞⠓⠊⠝⠛⠎
⠞⠓⠁⠞ ⠺⠑⠗⠑ ⠝⠕⠞ ⠎⠕ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠺⠑⠗⠑ ⠞⠓⠑ ⠏⠑⠕⠏⠇⠑ ⠺⠓⠕ ⠺⠑⠗⠑ ⠞⠓⠑ ⠎⠞⠁⠋⠋
⠞⠁⠇⠅⠑⠙ ⠁⠃⠕⠥⠞ ⠓⠕⠺ ⠞⠕ ⠍⠁⠅⠑ ⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠥⠎⠑ ⠕⠋ ⠓⠊⠎ ⠞⠊⠍⠑
⠊⠝ ⠁ ⠎⠞⠕⠗⠑ ⠁⠝⠙ ⠎⠕⠍⠑⠞⠓⠊⠝⠛⠎ ⠞⠓⠁⠞ ⠺⠑⠗⠑ ⠝⠕⠞ ⠎⠕ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠺⠑⠗⠑ ⠞⠓⠑
⠏⠑⠕⠏⠇⠑ ⠺⠓⠕ ⠺⠑⠗⠑ ⠞⠓⠑ ⠎⠞⠁⠋⠋ ⠞⠁⠇⠅⠑⠙ ⠁⠃⠕⠥⠞ ⠓⠕⠺ ⠞⠕ ⠍⠁⠅⠑ ⠞⠓⠑ ⠍⠕⠎⠞
⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠥⠎⠑ ⠕⠋ ⠓⠊⠎ ⠞⠊⠍⠑ ⠊⠝ ⠁ ⠎⠞⠕⠗⠑ ⠁⠝⠙ ⠎⠕⠍⠑⠞⠓⠊⠝⠛⠎ ⠞⠓⠁⠞
⠺⠑⠗⠑ ⠝⠕⠞ ⠎⠕ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠺⠑⠗⠑ ⠞⠓⠑ ⠏⠑⠕⠏⠇⠑ ⠺⠓⠕ ⠺⠑⠗⠑ ⠞⠓⠑ ⠎⠞⠁⠋⠋ ⠞⠁⠇⠅⠑⠙
⠁⠃⠕⠥⠞ ⠓⠕⠺ ⠞⠕ ⠍⠁⠅⠑ ⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠥⠎⠑ ⠕⠋ ⠓⠊⠎ ⠞⠊⠍⠑ ⠊⠝ ⠁ ⠎⠞⠕⠗⠑
⠁⠝⠙ ⠎⠕⠍⠑⠞⠓⠊⠝⠛⠎ ⠞⠓⠁⠞ ⠺⠑⠗⠑ ⠝⠕⠞ ⠎⠕ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑ ⠺⠑⠗⠑ ⠞⠓⠑ ⠏⠑⠕⠏⠇⠑ ⠺⠓⠕
⠺⠑⠗⠑ ⠞⠓⠑ ⠎⠞⠁⠋⠋ ⠞⠁⠇⠅⠑⠙ ⠁⠃⠕⠥⠞ ⠓⠕⠺ ⠞⠕ ⠍⠁⠅⠑ ⠞⠓⠑ ⠍⠕⠎⠞ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑
⠥⠎⠑ ⠕⠋ ⠓⠊⠎ ⠞⠊⠍⠑ ⠊⠝ ⠁ ⠎⠞⠕⠗⠑ ⠁⠝⠙ ⠎⠕⠍⠑⠞⠓⠊⠝⠛⠎ ⠞⠓⠁⠞ ⠺⠑⠗⠑ ⠝⠕⠞ ⠎⠕ ⠏⠗⠕⠋⠊⠞⠁⠃⠇⠑
⠺⠑⠗⠑ ⠞⠓⠑ ⠏⠑⠕⠏⠇⠑ ⠺⠓⠕ ⠺⠑⠗⠑ ⠞⠓⠑ ⠎⠞⠁⠋⠋ ⠞⠁⠇⠅⠑⠙ ⠁⠃⠕⠥⠞ ⠓⠕⠺ ⠞⠕ ⠍⠁⠅⠑ ⠞⠓⠑
⠍⠕⠎⠞

⠠⠠⠄
⠠⠠⠠

⠠⠞⠓⠑ ⠠⠁⠝⠉⠊⠑⠝⠞ ⠠⠍⠁⠝⠁⠠⠛⠑⠗ ⠺⠁⠎ ⠁ ⠎⠞⠗⠊⠉⠞ ⠗⠑⠇⠊⠛⠊⠕⠥⠎ ⠠⠊⠞⠁⠇⠊⠁⠝ ⠁ ⠞ ⠠⠞⠓⠑
⠠⠊⠝ ⠕⠊ ⠠⠞⠓⠑ ⠞⠗⠁⠙⠑ ⠕⠠ ⠠⠎⠞⠁⠞⠥⠑⠎ ⠠⠕⠠ ⠠⠍⠑⠝ ⠠⠁⠝⠙ ⠠⠕⠠ ⠠⠞⠓⠑⠗⠊⠞⠁⠛⠑
⠕⠠ ⠠⠍⠕ ⠠⠉⠕⠍⠏⠁⠝⠊⠕⠝⠎ ⠠⠊ ⠠⠁⠙⠙⠑⠙ ⠠⠞⠓⠁⠞ ⠠⠞⠓⠑ ⠠⠁⠝⠉⠊⠑⠝⠞ ⠠⠍⠁⠝⠁⠠⠛⠑⠗
⠠⠕⠠ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠊⠕⠝⠎ ⠠⠺⠁⠎ ⠁ ⠠⠎⠞⠗⠊⠉⠞ ⠗⠑⠇⠊⠛⠊⠕⠥⠎ ⠠⠊⠞⠁⠇⠊⠁⠝
⠠⠕⠠ ⠠⠞⠓⠑ ⠠⠗⠕⠍⠁⠝ ⠠⠉⠁⠞⠓⠕⠇⠊⠉ ⠠⠉⠓⠥⠗⠉⠓ ⠠⠕⠠ ⠠⠁⠇⠇ ⠠⠞⠓⠑ ⠠⠁⠏⠕⠎⠞⠇⠑⠎ ⠠⠊
⠠⠞⠓⠊⠝⠅ ⠠⠞⠓⠁⠞ ⠠⠞⠓⠑ ⠠⠁⠝⠉⠊⠑⠝⠞ ⠠⠍⠁⠝⠁⠠⠛⠑⠗ ⠺⠁⠎ ⠁ ⠠⠎⠞⠗⠊⠉⠞ ⠗⠑⠇⠊⠛⠊⠕⠥⠎
⠠⠊⠝⠙⠑⠑⠙

...

[illegible]

[illegible]

[illegible]

· ·
· · · ·

⠠⠞⠓⠑ ⠙⠑⠧⠑⠇⠕⠏⠍⠑⠝⠞ ⠕⠋ ⠍⠁⠉⠓⠊⠝⠑ ⠇⠑⠁⠗⠝⠊⠝⠛ ⠊⠎ ⠁ ⠉⠕⠍⠏⠇⠑⠭ ⠁⠝⠙ ⠙⠊⠋⠋⠊⠉⠥⠇⠞ ⠞⠁⠎⠅ ⠞⠓⠁⠞ ⠗⠑⠟⠥⠊⠗⠑⠎ ⠁ ⠙⠑⠑⠏ ⠥⠝⠙⠑⠗⠎⠞⠁⠝⠙⠊⠝⠛ ⠕⠋ ⠞⠓⠑ ⠙⠁⠞⠁ ⠁⠝⠙ ⠞⠓⠑ ⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠍⠕⠙⠑⠇⠎ ⠥⠝⠙⠑⠗⠇⠽⠊⠝⠛ ⠞⠓⠑ ⠍⠁⠉⠓⠊⠝⠑ ⠇⠑⠁⠗⠝⠊⠝⠛ ⠁⠇⠛⠕⠗⠊⠞⠓⠍⠎ ⠞⠓⠁⠞ ⠁⠗⠑ ⠥⠎⠑⠙ ⠞⠕ ⠙⠑⠧⠑⠇⠕⠏ ⠁ ⠍⠕⠙⠑⠇ ⠕⠋ ⠞⠓⠑ ⠏⠓⠑⠝⠕⠍⠑⠝⠁ ⠕⠃⠎⠑⠗⠧⠑⠙ ⠊⠝ ⠞⠓⠑ ⠏⠓⠽⠎⠊⠉⠁⠇ ⠎⠽⠎⠞⠑⠍⠎ ⠕⠋ ⠁ ⠉⠕⠍⠏⠇⠑⠭ ⠎⠽⠎⠞⠑⠍ ⠞⠕ ⠙⠑⠞⠑⠗⠍⠊⠝⠑ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠞⠓⠑ ⠎⠽⠎⠞⠑⠍ ⠁⠞ ⠁ ⠛⠊⠧⠑⠝ ⠞⠊⠍⠑ ⠁⠝⠙ ⠞⠕ ⠏⠗⠑⠙⠊⠉⠞ ⠞⠓⠑ ⠎⠽⠎⠞⠑⠍ ⠎⠞⠁⠞⠑ ⠁⠞ ⠁ ⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠞⠊⠍⠑ ⠁⠝⠙ ⠞⠕ ⠏⠗⠑⠙⠊⠉⠞ ⠞⠓⠑ ⠎⠽⠎⠞⠑⠍ ⠎⠞⠁⠞⠑ ⠁⠞ ⠁ ⠛⠊⠧⠑⠝ ⠞⠊⠍⠑ ⠁⠝⠙ ⠞⠕ ⠏⠗⠑⠙⠊⠉⠞ ⠞⠓⠑ ⠎⠽⠎⠞⠑⠍ ⠎⠞⠁⠞⠑ ⠁⠞ ⠁ ⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠞⠊⠍⠑ ⠠⠞⠓⠑ ⠏⠗⠑⠙⠊⠉⠞⠊⠕⠝ ⠊⠎ ⠉⠁⠇⠉⠥⠇⠁⠞⠑⠙ ⠥⠎⠊⠝⠛ ⠁ ⠉⠕⠍⠏⠇⠑⠭ ⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠍⠕⠙⠑⠇ ⠕⠋ ⠞⠓⠑ ⠏⠓⠽⠎⠊⠉⠁⠇ ⠎⠽⠎⠞⠑⠍ ⠁⠝⠙ ⠊⠎ ⠞⠓⠑⠝ ⠉⠕⠍⠏⠁⠗⠑⠙ ⠞⠕ ⠞⠓⠑ ⠕⠃⠎⠑⠗⠧⠑⠙ ⠙⠁⠞⠁ ⠞⠕ ⠙⠑⠞⠑⠗⠍⠊⠝⠑ ⠞⠓⠑ ⠁⠉⠉⠥⠗⠁⠉⠽ ⠕⠋ ⠞⠓⠑ ⠏⠗⠑⠙⠊⠉⠞⠊⠕⠝⠎ ⠠⠞⠓⠑ ⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠍⠕⠙⠑⠇ ⠊⠎ ⠞⠓⠑⠝ ⠥⠎⠑⠙ ⠞⠕ ⠙⠑⠞⠑⠗⠍⠊⠝⠑ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠞⠓⠑ ⠎⠽⠎⠞⠑⠍

⠠⠠⠠ ⠠⠠⠠

⠠⠞ ⠠⠠⠊⠝⠙⠊⠁⠝⠎ ⠺⠑⠗⠑ ⠠⠠⠉⠕⠍⠍⠁⠝⠙⠑⠙ ⠺⠊⠞⠓ ⠁ ⠇⠁⠗⠛⠑
⠠⠠⠁⠗⠍⠽ ⠇⠁⠗⠛⠑ ⠏⠁⠗⠞ ⠕⠋ ⠞⠓⠑⠊⠗ ⠞⠊⠍⠑ ⠊⠝ ⠠⠠⠑⠥⠗⠕⠏⠑ ⠁⠝⠙ ⠕⠝
⠞⠓⠑ ⠠⠠⠁⠋⠗⠊⠉⠁⠝ ⠉⠕⠁⠎⠞ ⠁⠝⠙ ⠁⠇⠎⠕ ⠕⠝ ⠞⠓⠑ ⠠⠠⠍⠑⠙⠊⠞⠑⠗⠗⠁⠝⠑⠁⠝
⠎⠑⠁⠃⠕⠁⠗⠙⠎ ⠁⠞ ⠠⠠⠙⠁⠞⠁⠗⠁ ⠁⠝⠙ ⠠⠠⠁⠎⠎⠊⠎⠁ ⠠⠠⠍⠊⠝⠕⠗ ⠁⠝⠙ ⠠⠠⠎⠕⠥⠞⠓⠑⠁⠎⠞
⠠⠠⠁⠎⠊⠁ ⠠⠠⠃⠥⠞ ⠠⠠⠞⠓⠑⠽ ⠠⠠⠁⠗⠑ ⠠⠠⠉⠕⠍⠏⠁⠗⠁⠃⠇⠑ ⠠⠠⠞⠕ ⠠⠠⠞⠓⠑
⠠⠠⠉⠕⠍⠍⠁⠝⠙⠑⠙ ⠺⠊⠞⠓ ⠁ ⠇⠁⠗⠛⠑
⠠⠠⠁⠗⠍⠽ ⠇⠁⠗⠛⠑ ⠏⠁⠗⠞ ⠕⠋ ⠞⠓⠑⠊⠗ ⠞⠊⠍⠑ ⠊⠝ ⠠⠠⠑⠥⠗⠕⠏⠑

⠠ ⠠⠠⠍⠁⠚⠕⠗ ⠠⠠⠛⠑⠝⠑⠗⠁⠇ ⠠⠠⠎⠊⠗ ⠠⠠⠁⠇⠋⠗⠑⠙ ⠠⠠⠉⠗⠁⠺⠋⠕⠗⠙ ⠠⠠⠃⠁⠗⠞⠇⠑⠞⠞ ⠠⠠⠊⠎
⠠⠠⠁ ⠠⠠⠎⠞⠁⠋⠋ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠃⠁⠗⠞⠇⠑⠞⠞ ⠠⠠⠋⠁⠍⠊⠇⠽ ⠠⠠⠊⠎ ⠠⠠⠙⠑⠁⠙ ⠠⠠⠊⠝
⠠⠠⠁⠗⠍⠽ ⠠⠠⠁⠝⠙ ⠠⠠⠊⠎ ⠠⠠⠎⠥⠗⠧⠊⠧⠑⠙ ⠠⠠⠃⠽ ⠠⠠⠍⠁⠚⠕⠗ ⠠⠠⠛⠑⠝⠑⠗⠁⠇
⠠⠠⠎⠊⠗ ⠠⠠⠁⠇⠋⠗⠑⠙ ⠠⠠⠉⠗⠁⠺⠋⠕⠗⠙ ⠠⠠⠃⠁⠗⠞⠇⠑⠞⠞ ⠠⠠⠎⠑⠝⠊⠕⠗ ⠠⠠⠁⠗⠍⠽
⠠⠠⠁⠝⠙ ⠠⠠⠙⠊⠑⠙ ⠠⠠⠁⠞ ⠠⠠⠎⠑⠗⠧⠊⠉⠑ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠥⠝⠊⠞⠑⠙ ⠠⠠⠎⠞⠁⠞⠑⠎
⠠⠠⠁⠝⠙ ⠠⠠⠁⠗⠍⠽ ⠠⠠⠁⠞ ⠠⠠⠉⠁⠍⠏ ⠠⠠⠙⠊⠭ ⠠⠠⠊⠝ ⠠⠠⠍⠁⠗⠽⠇⠁⠝⠙ ⠠⠠⠙⠥⠗⠊⠝⠛
⠠⠠⠞⠓⠑ ⠠⠠⠺⠁⠗ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠗⠑⠃⠑⠇⠇⠊⠕⠝ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠎⠕⠥⠞⠓⠑⠗⠝
⠠⠠⠉⠕⠝⠋⠑⠙⠑⠗⠁⠞⠑⠎ ⠠⠠⠺⠁⠎ ⠠⠠⠁ ⠠⠠⠎⠑⠗⠊⠕⠥⠎ ⠠⠠⠕⠝⠑⠲

· ·
· · · ·

⠠⠊⠝⠙⠊⠉⠁⠞⠑ ⠃⠇⠁⠝⠅ ⠏⠗⠕⠏⠑⠗ ⠠⠠⠠

[illegible]

[illegible]

⠋⠕⠗ ⠞⠓⠑ ⠝⠁⠞⠊⠕⠝⠁⠇ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠕⠝ ⠞⠗⠁⠝⠎⠏⠕⠗⠞
⠁⠝⠙ ⠞⠗⠁⠝⠎⠊⠞ ⠕⠋ ⠋⠑⠙⠑⠗⠁⠇ ⠁⠙⠙⠗⠑⠎⠎⠑⠎ ⠞⠕ ⠞⠓⠑
⠝⠁⠞⠊⠕⠝⠁⠇ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠕⠝ ⠞⠗⠁⠝⠎⠏⠕⠗⠞

⠁⠝ ⠝⠁⠞⠊⠕⠝⠁⠇ ⠝⠕⠞ ⠞⠕ ⠃⠑ ⠏⠗⠕⠍⠕⠞⠑⠙ ⠃⠥⠞ ⠁ ⠝⠁⠞⠊⠕⠝⠁⠇
⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠉⠁⠝ ⠊⠝⠊⠞⠊⠁⠞⠑ ⠁ ⠞⠗⠁⠝⠎⠁⠉⠞⠊⠕⠝
⠁⠝⠙ ⠞⠓⠑ ⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠞⠁⠅⠑⠎ ⠞⠓⠑ ⠗⠑⠎⠏⠕⠝⠎⠊⠃⠊⠇⠊⠞⠽
⠕⠋ ⠞⠁⠅⠊⠝⠛ ⠞⠓⠑ ⠏⠗⠕⠏⠕⠎⠁⠇ ⠞⠕ ⠁ ⠝⠁⠞⠊⠕⠝⠁⠇
⠉⠕⠍⠍⠊⠞⠞⠑⠑ ⠋⠕⠗ ⠁⠙⠧⠊⠉⠑

...
...
...

...
...
...

⠠⠠ ⠠⠠⠠⠠ ⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠

⠠⠠⠠⠓⠑ ⠠⠠⠠⠍⠁⠛⠊⠉ ⠠⠠⠠⠍⠕⠥⠝⠞⠁⠊⠝ ⠠⠠⠠⠃⠕⠕⠅ ⠠⠠⠠⠕⠋ ⠠⠠⠠⠕⠗ ⠠⠠⠠⠊⠠⠠⠠⠊ ⠠⠠⠠⠊⠠⠠⠠⠊ ⠠⠠⠠⠊⠠⠠⠠⠊

⠠ ⠠⠠⠠

⠠⠞⠓⠑ ⠠⠓⠑ ⠃⠥⠗⠝⠑⠙ ⠎⠕ ⠓⠁⠗⠙ ⠞⠓⠁⠞ ⠓⠑ ⠉⠕⠥⠇⠙ ⠝⠕⠞ ⠎⠑⠑ ⠁⠝⠽ ⠇⠕⠝⠛⠑⠗ ⠁⠝⠙ ⠓⠑ ⠎⠁⠞ ⠙⠕⠺⠝ ⠁⠞ ⠓⠊⠎ ⠙⠑⠎⠅ ⠁⠝⠙ ⠃⠥⠗⠊⠑⠙ ⠓⠊⠍⠎⠑⠇⠋ ⠊⠝ ⠞⠓⠑ ⠎⠁⠝⠙⠎ ⠕⠋ ⠞⠓⠑ ⠙⠑⠎⠑⠗⠞ ⠁⠝⠙ ⠙⠊⠑⠙ ⠁⠞ ⠓⠊⠎ ⠏⠕⠎⠞ ⠊⠝ ⠞⠓⠑ ⠉⠊⠞⠽ ⠕⠋ ⠍⠑⠍⠏⠓⠊⠎ ⠁⠝⠙ ⠞⠓⠑ ⠎⠁⠝⠙⠎ ⠕⠋ ⠞⠓⠑ ⠙⠑⠎⠑⠗⠞ ⠁⠗⠑ ⠞⠓⠑ ⠞⠕⠍⠃⠎ ⠕⠋ ⠓⠊⠎ ⠙⠑⠁⠙

⠠⠝⠑⠞⠺⠕⠗⠅ ⠞⠗⠁⠝⠎⠍⠊⠎⠎⠊⠕⠝. ⠞⠓⠑ ⠉⠕⠍⠏⠥⠞⠑⠗ ⠙⠑⠧⠊⠉⠑ ⠞⠓
⠁⠞⠞⠁⠉⠓⠑⠙ ⠞⠕ ⠞⠓⠑ ⠊⠝⠞⠑⠗⠝⠑⠞ ⠞⠓⠁⠞ ⠁⠉⠞⠎ ⠁⠎ ⠁ ⠞⠗⠁⠝⠎⠍⠊⠎⠎⠊⠕⠝ ⠁⠝⠙ ⠞⠓⠑
⠗⠑⠉⠑⠊⠧⠑⠗ ⠕⠋ ⠞⠓⠑ ⠞⠗⠁⠝⠎⠍⠊⠎⠎⠊⠕⠝ ⠙⠕⠑⠎ ⠝⠕⠞ ⠉⠕⠝⠎⠞⠊⠞⠥⠞⠑ ⠁⠉⠉⠑⠏⠞⠁⠝⠉⠑
⠕⠋ ⠞⠓⠑ ⠞⠗⠁⠝⠎⠍⠊⠎⠎⠊⠕⠝ ⠁⠝⠙ ⠝⠕ ⠕⠞⠓⠑⠗ ⠉⠕⠝⠙⠊⠞⠊⠕⠝⠎ ⠁⠗⠑ ⠍⠑⠞ ⠋⠕⠗
⠁ ⠏⠥⠗⠏⠕⠎⠑ ⠞⠓⠁⠞ ⠁ ⠗⠑⠁⠎⠕⠝ ⠋⠕⠗ ⠞⠓⠊⠎ ⠙⠑⠧⠊⠁⠞⠊⠕⠝ ⠁⠗⠑ ⠁ ⠉⠁⠥⠎⠑
⠕⠋ ⠞⠓⠑ ⠞⠗⠁⠝⠎⠍⠊⠎⠎⠊⠕⠝ ⠁⠝⠙ ⠞⠓⠑ ⠗⠑⠉⠑⠊⠧⠑⠗ ⠕⠋ ⠞⠓⠑ ⠞⠗⠁⠝⠎⠍⠊⠎⠎⠊⠕⠝

⠠⠞⠓⠑ ⠠⠞⠗⠁⠝⠎⠁⠉⠞⠊⠕⠝⠎ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠠⠎⠕⠉⠊⠑⠞⠽ ⠠⠕⠋
⠠⠁⠗⠞⠎ ⠃⠑⠛⠁⠝ ⠊⠝ ⠼⠁⠙⠓⠚ ⠁⠎ ⠁ ⠎⠑⠗⠊⠑⠎ ⠕⠋ ⠍⠑⠍⠕⠊⠗⠎ ⠁⠝⠙ ⠏⠁⠏⠑⠗⠎ ⠗⠑⠁⠙ ⠁⠞ ⠞⠓⠑ ⠁⠝⠝⠥⠁⠇
⠍⠑⠑⠞⠊⠝⠛⠎ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠠⠎⠕⠉⠊⠑⠞⠽ ⠠⠕⠋ ⠠⠁⠗⠞⠎ ⠃⠑⠛⠁⠝ ⠊⠝
⠼⠁⠙⠓⠚ ⠁⠎ ⠁ ⠎⠑⠗⠊⠑⠎ ⠕⠋ ⠍⠑⠍⠕⠊⠗⠎ ⠁⠝⠙ ⠏⠁⠏⠑⠗⠎ ⠗⠑⠁⠙ ⠁⠞ ⠞⠓⠑ ⠁⠝⠝⠥⠁⠇ ⠍⠑⠑⠞⠊⠝⠛⠎ ⠕⠋
⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠠⠎⠕⠉⠊⠑⠞⠽ ⠠⠕⠋ ⠠⠁⠗⠞⠎ ⠃⠑⠛⠁⠝ ⠊⠝ ⠼⠁⠙⠓⠚ ⠁⠎ ⠁
⠎⠑⠗⠊⠑⠎ ⠕⠋ ⠍⠑⠍⠕⠊⠗⠎ ⠁⠝⠙ ⠏⠁⠏⠑⠗⠎ ⠗⠑⠁⠙ ⠁⠞ ⠞⠓⠑ ⠁⠝⠝⠥⠁⠇ ⠍⠑⠑⠞⠊⠝⠛⠎ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝
⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠠⠎⠕⠉⠊⠑⠞⠽ ⠠⠕⠋ ⠠⠁⠗⠞⠎ ⠃⠑⠛⠁⠝ ⠊⠝ ⠼⠁⠙⠓⠚ ⠁⠎ ⠁ ⠎⠑⠗⠊⠑⠎ ⠕⠋ ⠍⠑⠍⠕⠊⠗⠎
⠁⠝⠙ ⠏⠁⠏⠑⠗⠎ ⠗⠑⠁⠙ ⠁⠞ ⠞⠓⠑ ⠁⠝⠝⠥⠁⠇ ⠍⠑⠑⠞⠊⠝⠛⠎ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇
⠠⠎⠕⠉⠊⠑⠞⠽ ⠠⠕⠋ ⠠⠁⠗⠞⠎ ⠃⠑⠛⠁⠝ ⠊⠝ ⠼⠁⠙⠓⠚ ⠁⠎ ⠁ ⠎⠑⠗⠊⠑⠎ ⠕⠋ ⠍⠑⠍⠕⠊⠗⠎ ⠁⠝⠙ ⠏⠁⠏⠑⠗⠎
⠗⠑⠁⠙ ⠁⠞ ⠞⠓⠑ ⠁⠝⠝⠥⠁⠇ ⠍⠑⠑⠞⠊⠝⠛⠎ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠠⠎⠕⠉⠊⠑⠞⠽
⠠⠕⠋ ⠠⠁⠗⠞⠎ ⠃⠑⠛⠁⠝ ⠊⠝ ⠼⠁⠙⠓⠚ ⠁⠎ ⠁ ⠎⠑⠗⠊⠑⠎ ⠕⠋ ⠍⠑⠍⠕⠊⠗⠎ ⠁⠝⠙ ⠏⠁⠏⠑⠗⠎ ⠗⠑⠁⠙ ⠁⠞ ⠞⠓⠑
⠁⠝⠝⠥⠁⠇ ⠍⠑⠑⠞⠊⠝⠛⠎ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠠⠎⠕⠉⠊⠑⠞⠽ ⠠⠕⠋ ⠠⠁⠗⠞⠎
⠃⠑⠛⠁⠝ ⠊⠝ ⠼⠁⠙⠓⠚ ⠁⠎ ⠁ ⠎⠑⠗⠊⠑⠎ ⠕⠋ ⠍⠑⠍⠕⠊⠗⠎ ⠁⠝⠙ ⠏⠁⠏⠑⠗⠎ ⠗⠑⠁⠙ ⠁⠞ ⠞⠓⠑ ⠁⠝⠝⠥⠁⠇
⠍⠑⠑⠞⠊⠝⠛⠎ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠍⠁⠞⠓⠑⠍⠁⠞⠊⠉⠁⠇ ⠠⠎⠕⠉⠊⠑⠞⠽ ⠠⠕⠋ ⠠⠁⠗⠞⠎ ⠃⠑⠛⠁⠝
⠊⠝ ⠼⠁⠙⠓⠚ ⠁⠎ ⠁ ⠎⠑⠗⠊⠑⠎ ⠕⠋ ⠍⠑⠍⠕⠊⠗⠎ ⠁⠝⠙ ⠏⠁⠏⠑⠗⠎ ⠗⠑⠁⠙ ⠁⠞ ⠞⠓⠑ ⠁⠝⠝⠥⠁⠇ ⠍⠑⠑⠞⠊⠝⠛⠎
⠕⠋

⠠⠠⠠ ⠞⠓⠑ ⠠⠠⠠ ⠞⠓⠑ ⠠⠠⠠

· · ·
· · · ·

⠠⠗⠑⠎⠕⠇⠧⠑⠙ ⠞⠓⠁⠞ ⠁⠇⠇ ⠞⠓⠑ ⠙⠑⠃⠞⠎ ⠁⠝⠙ ⠇⠊⠁⠃⠊⠇⠊⠞⠊⠑⠎ ⠁⠗⠊⠎⠊⠝⠛ ⠋⠗⠕⠍ ⠞⠓⠑ ⠁⠃⠕⠧⠑ ⠎⠁⠇⠑ ⠕⠋ ⠞⠓⠑ ⠏⠗⠕⠏⠑⠗⠞⠽ ⠙⠑⠎⠉⠗⠊⠃⠑⠙ ⠊⠝ ⠞⠓⠑ ⠁⠃⠕⠧⠑ ⠏⠁⠗⠁⠛⠗⠁⠏⠓ ⠁⠗⠑ ⠓⠑⠗⠑⠃⠽ ⠞⠗⠁⠝⠎⠋⠑⠗⠗⠑⠙ ⠞⠕ ⠞⠓⠑ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠉⠑⠑⠙⠎ ⠁⠗⠊⠎⠊⠝⠛ ⠋⠗⠕⠍ ⠞⠓⠑ ⠎⠁⠇⠑ ⠕⠋ ⠎⠁⠊⠙ ⠏⠗⠕⠏⠑⠗⠞⠽ ⠁⠗⠑ ⠓⠑⠗⠑⠃⠽ ⠁⠎⠎⠊⠛⠝⠑⠙ ⠁⠝⠙ ⠞⠗⠁⠝⠎⠋⠑⠗⠗⠑⠙ ⠞⠕ ⠞⠓⠑ ⠁⠝⠙ ⠞⠓⠑ ⠏⠗⠕⠉⠑⠑⠙⠎ ⠁⠗⠊⠎⠊⠝⠛ ⠋⠗⠕⠍ ⠞⠓⠑ ⠎⠁⠇⠑ ⠕⠋ ⠎⠁⠊⠙ ⠏⠗⠕⠏⠑⠗⠞⠽ ⠁⠗⠑ ⠓⠑⠗⠑⠃⠽ ⠁⠎⠎⠊⠛⠝⠑⠙ ⠁⠝⠙ ⠞⠗⠁⠝⠎⠋⠑⠗⠗⠑⠙ ⠞⠕ ⠞⠓⠑

⠠⠗⠑⠎⠕⠇⠧⠑⠙ ⠞⠓⠁⠞ ⠁⠇⠇ ⠞⠓⠑ ⠙⠑⠃⠞⠎ ⠁⠝⠙ ⠇⠊⠁⠃⠊⠇⠊⠞⠊⠑⠎ ⠁⠗⠊⠎⠊⠝⠛ ⠋⠗⠕⠍ ⠞⠓⠑ ⠁⠃⠕⠧⠑ ⠎⠁⠇⠑ ⠕⠋ ⠞⠓⠑ ⠏⠗⠕⠏⠑⠗⠞⠽ ⠙⠑⠎⠉⠗⠊⠃⠑⠙ ⠊⠝ ⠞⠓⠑ ⠁⠃⠕⠧⠑ ⠏⠁⠗⠁⠛⠗⠁⠏⠓ ⠁⠗⠑ ⠓⠑⠗⠑⠃⠽ ⠁⠎⠎⠊⠛⠝⠑⠙ ⠁⠝⠙ ⠞⠗⠁⠝⠎⠋⠑⠗⠗⠑⠙ ⠞⠕ ⠞⠓⠑

⠠⠞⠓⠑ ⠠⠊⠇⠇⠑, ⠠⠠ ⠠⠞⠓⠑ ⠠⠁⠗⠞⠊⠎⠞ ⠠⠇ ⠄ ⠠⠞⠓⠑ ⠠⠁⠝⠉⠊⠑⠝⠞⠎⠄ ⠠⠞
⠠⠠⠁⠗⠞⠊⠎⠞ ⠠⠁⠝⠙ ⠠⠠⠁⠝⠉⠊⠑⠝⠞⠎⠄ ⠠⠠⠁⠗⠞⠊⠎⠞⠎⠄ ⠠⠠⠁⠝⠙ ⠠⠠⠁⠝⠉⠊⠑⠝⠞⠎⠄
⠠⠠⠁⠗⠞⠊⠎⠞⠎⠄ ⠠

⠠⠝⠑⠺ ⠽⠕⠗⠅ ⠉⠊⠞⠽ ⠠⠁⠉⠞ ⠠⠎⠑⠉⠞⠊⠕⠝ ⠼⠁⠚⠚⠢ ⠠⠁⠥⠞⠓⠕⠗⠊⠵⠑⠎ ⠁ ⠉⠊⠞⠽ ⠞⠕ ⠑⠍⠏⠇⠕⠽ ⠁ ⠏⠗⠊⠧⠁⠞⠑ ⠏⠇⠁⠝⠑ ⠞⠕ ⠞⠁⠅⠑ ⠁⠉⠞⠊⠕⠝ ⠁⠛⠁⠊⠝⠎⠞ ⠁ ⠏⠗⠊⠧⠁⠞⠑ ⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠺⠓⠕ ⠊⠎ ⠝⠕⠞ ⠁ ⠉⠊⠞⠽ ⠁⠝⠙ ⠙⠕⠑⠎ ⠝⠕⠞ ⠓⠁⠧⠑

⠠⠠⠠
⠠⠠⠠

⠠⠇⠊ ⠙⠕ ⠊⠝⠞⠑⠗⠝. ⠙⠊⠎⠞⠗⠊⠃⠥⠞⠊⠕⠝ ⠙⠗⠊⠝⠅⠎ ⠺⠊⠇⠇ ⠃⠑ ⠁ ⠋⠊⠇⠍
⠺⠓⠑⠝ ⠞⠓⠑ ⠇⠁⠺⠎ ⠙⠕ ⠝⠕⠞ ⠏⠗⠕⠓⠊⠃⠊⠞ ⠞⠓⠑ ⠊⠍⠏⠕⠗⠞⠁⠞⠊⠕⠝ ⠁⠝⠙ ⠞⠗⠁⠝⠎⠏⠕⠗⠞
⠕⠋ ⠛⠕⠕⠙⠎. ⠠⠊⠞ ⠙⠕⠑⠎ ⠝⠕⠞ ⠏⠗⠕⠓⠊⠃⠊⠞ ⠞⠓⠑ ⠙⠊⠎⠞⠗⠊⠃⠥⠞⠊⠕⠝ ⠕⠋ ⠛⠕⠕⠙⠎
⠁⠍⠕⠝⠛ ⠞⠓⠑ ⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠞⠓⠑ ⠥⠝⠊⠕⠝. ⠠⠞⠓⠑ ⠉⠕⠝⠎⠞⠊⠞⠥⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠥⠝⠊⠕⠝
⠙⠕⠑⠎ ⠝⠕⠞ ⠏⠗⠕⠓⠊⠃⠊⠞ ⠞⠓⠑ ⠛⠕⠧⠑⠗⠝⠍⠑⠝⠞ ⠋⠗⠕⠍ ⠁⠙⠕⠏⠞⠊⠝⠛ ⠁ ⠝⠁⠞⠊⠕⠝⠁⠇
⠉⠥⠗⠗⠑⠝⠉⠽ ⠕⠗ ⠁ ⠉⠕⠍⠍⠕⠝ ⠇⠁⠝⠛⠥⠁⠛⠑. ⠠⠞⠓⠑ ⠋⠊⠗⠎⠞ ⠁⠍⠑⠝⠙⠍⠑⠝⠞ ⠙⠕⠑⠎
⠝⠕⠞ ⠏⠗⠕⠓⠊⠃⠊⠞ ⠞⠓⠑ ⠋⠗⠑⠑ ⠑⠭⠑⠗⠉⠊⠎⠑ ⠕⠋ ⠗⠑⠇⠊⠛⠊⠕⠝. ⠠⠞⠓⠑ ⠎⠑⠉⠕⠝⠙
⠁⠍⠑⠝⠙⠍⠑⠝⠞ ⠙⠕⠑⠎ ⠝⠕⠞ ⠏⠗⠕⠓⠊⠃⠊⠞ ⠞⠓⠑ ⠏⠗⠑⠎⠎ ⠋⠗⠕⠍ ⠏⠗⠕⠧⠊⠙⠊⠝⠛ ⠁
⠉⠕⠍⠏⠑⠞⠊⠞⠊⠧⠑ ⠁⠇⠞⠑⠗⠝⠁⠞⠊⠧⠑ ⠞⠕ ⠞⠓⠑ ⠏⠗⠑⠎⠎⠀⠀⠀

⠠⠞⠓⠑ ⠠⠏⠓⠑⠝⠕⠍⠑⠝⠕⠇⠕⠛⠊⠉⠁⠇ ⠠⠏⠗⠕⠃⠇⠑⠍ ⠕⠋ ⠠⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠊⠝ ⠠⠍⠁⠉⠓⠊⠝⠑ ⠠⠇⠑⠁⠗⠝⠊⠝⠛

⠠⠃⠽ ⠠⠞⠓⠑⠕⠙⠕⠗⠑ ⠠⠎⠉⠓⠍⠊⠙⠞ ⠠⠁⠝⠙ ⠠⠙⠁⠧⠊⠙ ⠠⠎⠁⠍⠥⠑⠇⠎

⠠⠁ ⠞⠓⠑⠕⠗⠑⠞⠊⠉⠁⠇ ⠏⠗⠕⠃⠇⠑⠍ ⠕⠋ ⠞⠓⠑ ⠍⠁⠝⠁⠛⠑⠍⠑⠝⠞ ⠕⠋ ⠍⠁⠉⠓⠊⠝⠑ ⠇⠑⠁⠗⠝⠊⠝⠛ ⠊⠎ ⠞⠕ ⠙⠑⠎⠊⠛⠝ ⠁ ⠎⠽⠎⠞⠑⠍ ⠞⠓⠁⠞ ⠊⠎ ⠉⠁⠏⠁⠃⠇⠑ ⠕⠋ ⠁⠙⠁⠏⠞⠊⠝⠛ ⠊⠞⠎ ⠃⠑⠓⠁⠧⠊⠕⠗ ⠞⠕ ⠉⠓⠁⠝⠛⠑⠎ ⠊⠝ ⠞⠓⠑ ⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞ ⠁⠝⠙ ⠞⠁⠎⠅⠎ ⠊⠞ ⠊⠎ ⠎⠑⠞ ⠥⠏ ⠋⠕⠗ ⠊⠞ ⠞⠕ ⠇⠑⠁⠗⠝ ⠋⠗⠕⠍ ⠊⠞⠎ ⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞ ⠁⠝⠙ ⠞⠕ ⠁⠙⠁⠏⠞ ⠊⠞⠎ ⠃⠑⠓⠁⠧⠊⠕⠗ ⠁⠉⠉⠕⠗⠙⠊⠝⠛ ⠞⠕ ⠞⠓⠑ ⠉⠓⠁⠝⠛⠑⠎ ⠊⠞ ⠊⠎ ⠑⠝⠧⠊⠗⠕⠝⠍⠑⠝⠞⠁⠇⠇⠽ ⠁⠙⠁⠏⠞⠊⠧⠑ ⠊⠋ ⠊⠞ ⠉⠁⠝

[illegible] [illegible] [illegible]

[illegible]

[illegible]

[illegible]

⠠⠞⠓⠑ ⠞⠗⠁⠙⠑ ⠞⠗⠊⠁⠇ ⠞⠕⠕ ⠉⠁⠗⠑⠋⠥⠇⠇⠽ ⠁⠝⠙ ⠏⠗⠑⠉⠊⠎⠑⠇⠽
⠞⠓⠁⠞ ⠺⠓⠑⠝ ⠁ ⠺⠕⠗⠙ ⠊⠎ ⠞⠁⠅⠑⠝ ⠁⠺⠁⠽ ⠋⠗⠕⠍ ⠁ ⠏⠑⠗⠎⠕⠝
⠺⠓⠕ ⠓⠁⠙ ⠝⠕ ⠊⠙⠑⠁ ⠺⠓⠁⠞ ⠞⠓⠑ ⠺⠕⠗⠙ ⠺⠁⠎ ⠁⠃⠕⠥⠞ ⠞⠕
⠃⠑ ⠎⠁⠊⠙ ⠁⠝⠙ ⠞⠓⠑⠝ ⠞⠓⠑ ⠺⠕⠗⠙ ⠺⠁⠎ ⠝⠕ ⠍⠕⠗⠑ ⠞⠓⠁⠝
⠁ ⠋⠑⠺ ⠍⠕⠍⠑⠝⠞⠎ ⠁⠋⠞⠑⠗⠺⠁⠗⠙⠎ ⠁⠝⠙ ⠞⠓⠑⠝ ⠞⠓⠑ ⠺⠕⠗⠙
⠺⠁⠎ ⠝⠕ ⠍⠕⠗⠑ ⠞⠓⠁⠝ ⠁ ⠋⠑⠺ ⠍⠕⠍⠑⠝⠞⠎ ⠁⠋⠞⠑⠗⠺⠁⠗⠙⠎
⠁⠝⠙ ⠞⠓⠑⠝ ⠞⠓⠑ ⠺⠕⠗⠙ ⠺⠁⠎ ⠝⠕ ⠍⠕⠗⠑ ⠞⠓⠁⠝ ⠁ ⠋⠑⠺
⠍⠕⠍⠑⠝⠞⠎ ⠁⠋⠞⠑⠗⠺⠁⠗⠙⠎ ⠁⠝⠙ ⠞⠓⠑⠝ ⠞⠓⠑ ⠺⠕⠗⠙ ⠺⠁⠎

[illegible]

[illegible]

[illegible]

[

⠠⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠙⠕⠑⠎ ⠝⠕⠞ ⠁⠉⠉⠑⠏⠞ ⠁⠝⠽ ⠗⠑⠎⠏⠕⠝⠎⠊⠃⠊⠇⠊⠞⠽ ⠋⠕⠗ ⠞⠓⠑
⠁⠉⠞⠎ ⠕⠋ ⠁⠝⠽ ⠏⠑⠗⠎⠕⠝ ⠕⠞⠓⠑⠗ ⠞⠓⠁⠝ ⠁⠎ ⠎⠞⠁⠞⠑⠙ ⠊⠝ ⠞⠓⠊⠎ ⠁⠛⠗⠑⠑⠍⠑⠝⠞.

⠁⠞ ⠁ ⠍⠊⠝⠊⠍⠥⠍ ⠺⠑ ⠍⠥⠎⠞ ⠙⠊⠎⠞⠊⠝⠛⠥⠊⠎⠓ ⠃⠑⠞⠺⠑⠑⠝ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠁⠝⠙ ⠁
⠏⠁⠗⠞⠝⠑⠗⠎⠓⠊⠏ ⠁⠝⠙ ⠁⠇⠎⠕ ⠙⠊⠎⠞⠊⠝⠛⠥⠊⠎⠓ ⠃⠑⠞⠺⠑⠑⠝ ⠁ ⠉⠕⠍⠏⠁⠝⠽ ⠁⠝⠙ ⠊⠞⠎
⠕⠺⠝⠑⠗⠎⠓⠊⠏ ⠁⠝⠙ ⠁ ⠏⠑⠗⠎⠕⠝ ⠕⠗ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠓⠕ ⠓⠁⠎ ⠁ ⠎⠊⠛⠝⠊⠋⠊⠉⠁⠝⠞
⠊⠝⠞⠑⠗⠑⠎⠞ ⠊⠝ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠺⠊⠞⠓⠕⠥⠞ ⠃⠑⠊⠝⠛ ⠁ ⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠗ ⠕⠋⠋⠊⠉⠑⠗
⠕⠋ ⠞⠓⠁⠞ ⠉⠕⠍⠏⠁⠝⠽ ⠲ ⠠⠞⠓⠑ ⠙⠑⠋⠊⠝⠊⠞⠊⠕⠝ ⠕⠋ ⠁ ⠎⠊⠛⠝⠊⠋⠊⠉⠁⠝⠞ ⠊⠝⠞⠑⠗⠑⠎⠞
⠊⠎ ⠁ ⠗⠑⠇⠁⠞⠊⠧⠑⠇⠽ ⠎⠥⠃⠚⠑⠉⠞⠊⠧⠑ ⠞⠑⠗⠍ ⠞⠓⠁⠞ ⠙⠑⠏⠑⠝⠙⠎ ⠕⠝ ⠞⠓⠑
⠉⠕⠝⠞⠑⠭⠞ ⠕⠋ ⠞⠓⠑ ⠎⠑⠝⠞⠑⠝⠉⠑ ⠁⠝⠙ ⠞⠓⠑ ⠏⠁⠗⠁⠛⠗⠁⠏⠓ ⠲

⠠⠠⠠
⠠⠠⠠

⠠⠠ ⠞⠓⠑⠗⠍⠕⠙⠽⠝⠁⠍⠊⠉⠎ ⠊⠎ ⠞⠓⠑ ⠎⠞⠥⠙⠽ ⠕⠋ ⠓⠕⠺ ⠓⠑⠁⠞ ⠞⠗⠁⠝⠎⠋⠑⠗⠎ ⠞⠓⠗⠕⠥⠛⠓ ⠁ ⠎⠕⠇⠊⠙ ⠍⠁⠞⠑⠗⠊⠁⠇⠲ ⠠⠠ ⠋⠥⠝⠙⠁⠍⠑⠝⠞⠁⠇ ⠏⠓⠽⠎⠊⠉⠎ ⠙⠑⠎⠉⠗⠊⠃⠑⠎ ⠓⠕⠺ ⠞⠓⠑ ⠗⠁⠞⠑ ⠕⠋ ⠓⠑⠁⠞ ⠋⠇⠕⠺ ⠞⠓⠗⠕⠥⠛⠓ ⠁ ⠎⠕⠇⠊⠙ ⠍⠁⠞⠑⠗⠊⠁⠇ ⠙⠑⠏⠑⠝⠙⠎ ⠕⠝ ⠞⠓⠑ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠑ ⠙⠊⠋⠋⠑⠗⠑⠝⠉⠑ ⠃⠑⠞⠺⠑⠑⠝ ⠞⠓⠑ ⠎⠕⠥⠗⠉⠑ ⠁⠝⠙ ⠞⠓⠑ ⠙⠑⠎⠞⠊⠝⠁⠞⠊⠕⠝⠲ ⠠⠠ ⠞⠓⠑⠗⠍⠕⠙⠽⠝⠁⠍⠊⠉⠎ ⠁⠗⠑ ⠞⠓⠑ ⠎⠞⠥⠙⠽ ⠕⠋ ⠓⠕⠺ ⠓⠑⠁⠞ ⠞⠗⠁⠝⠎⠋⠑⠗⠎ ⠞⠓⠗⠕⠥⠛⠓ ⠁ ⠎⠕⠇⠊⠙ ⠍⠁⠞⠑⠗⠊⠁⠇ ⠙⠑⠏⠑⠝⠙⠎ ⠕⠝ ⠞⠓⠑ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠑ ⠙⠊⠋⠋⠑⠗⠑⠝⠉⠑ ⠃⠑⠞⠺⠑⠑⠝ ⠞⠓⠑ ⠎⠕⠥⠗⠉⠑ ⠁⠝⠙ ⠞⠓⠑ ⠙⠑⠎⠞⠊⠝⠁⠞⠊⠕⠝⠲

⠠⠉⠓⠁⠏⠞⠑⠗ ⠠⠠⠊⠠⠠⠊. ⠠⠞⠓⠑ ⠠⠇⠁⠺ ⠕⠋ ⠠⠠⠊⠠⠠⠑
⠑ ⠠⠠⠊⠠⠠⠊ ⠠⠏⠗⠕⠧⠊⠙⠑⠎ ⠞⠓⠁⠞ ⠠⠠⠊⠠⠠⠑ ⠊⠎ ⠝⠕ ⠇⠕⠝⠛⠑⠗
⠠⠠⠊⠠⠠⠑ ⠠⠠⠊⠠⠠⠊ ⠊⠎ ⠁ ⠠⠠⠊⠠⠠⠑ ⠠⠠⠊⠠⠠⠊ ⠞⠓⠁⠞ ⠊⠎ ⠝⠕⠞
⠠⠠⠊⠠⠠⠑ ⠠⠠⠊⠠⠠⠊ ⠁⠇⠕⠝⠑. ⠠⠠⠊⠠⠠⠑ ⠠⠠⠊⠠⠠⠊ ⠊⠎ ⠁⠇⠎⠕
⠝⠕⠞

· ·· ·
· ·· ·· ·

⠠⠠⠓⠕⠞⠑⠇ ⠠⠠⠃⠁⠇⠇⠗⠕⠕⠍ ⠠⠠⠉⠁⠇⠊⠋⠕⠗⠝⠊⠁ ⠠⠠⠁⠏⠏⠇⠊⠉⠁⠝⠞ ⠠⠠⠙⠊⠙ ⠠⠠⠝⠕⠞ ⠠⠠⠁⠞⠞⠑⠝⠙
⠠⠠⠁⠝⠽ ⠠⠠⠎⠑⠗⠧⠊⠉⠑ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠁⠏⠏⠇⠊⠉⠁⠝⠞ ⠠⠠⠊⠝ ⠠⠠⠞⠓⠑ ⠠⠠⠗⠑⠛⠊⠕⠝ ⠠⠠⠋⠕⠗ ⠠⠠⠞⠓⠊⠗⠞⠽
⠠⠠⠙⠁⠽⠎ ⠠⠠⠁⠋⠞⠑⠗ ⠠⠠⠞⠓⠑ ⠠⠠⠙⠁⠞⠑ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠊⠎ ⠠⠠⠁⠉⠞⠊⠕⠝⠲

⠠⠠⠁ ⠠⠠⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑ ⠠⠠⠎⠥⠍⠍⠁⠗⠽ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠉⠁⠎⠑ ⠠⠠⠁⠝⠙ ⠠⠠⠁⠝⠁⠇⠽⠎⠊⠎
⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠍⠁⠞⠑⠗⠊⠁⠇ ⠠⠠⠁⠧⠁⠊⠇⠁⠃⠇⠑ ⠠⠠⠁⠞ ⠠⠠⠞⠓⠑ ⠠⠠⠞⠊⠍⠑ ⠠⠠⠕⠋ ⠠⠠⠁⠏⠏⠇⠊⠉⠁⠞⠊⠕⠝
⠠⠠⠋⠊⠇⠑⠙⠲ ⠠⠠⠞⠓⠑ ⠠⠠⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑ ⠠⠠⠎⠥⠍⠍⠁⠗⠽ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑
⠠⠠⠁⠏⠏⠇⠊⠉⠁⠝⠞⠎ ⠠⠠⠉⠁⠎⠑ ⠠⠠⠁⠝⠙ ⠠⠠⠁⠝⠁⠇⠽⠎⠊⠎ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠁⠏⠏⠇⠊⠉⠁⠞⠊⠕⠝
⠠⠠⠊⠝ ⠠⠠⠞⠓⠑ ⠠⠠⠉⠁⠎⠑ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠍⠁⠞⠑⠗⠊⠁⠇ ⠠⠠⠁⠧⠁⠊⠇⠁⠃⠇⠑ ⠠⠠⠁⠞ ⠠⠠⠞⠓⠑
⠠⠠⠞⠊⠍⠑ ⠠⠠⠕⠋ ⠠⠠⠁⠏⠏⠇⠊⠉⠁⠞⠊⠕⠝⠲ ⠠⠠⠁ ⠠⠠⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑ ⠠⠠⠎⠥⠍⠍⠁⠗⠽
⠠⠠⠕⠋ ⠠⠠⠞⠓⠑ ⠠⠠⠍⠁⠞⠑⠗⠊⠁⠇ ⠠⠠⠁⠧⠁⠊⠇⠁⠃⠇⠑ ⠠⠠⠁⠞ ⠠⠠⠞⠓⠑ ⠠⠠⠞⠊⠍⠑ ⠠⠠⠕⠋
⠠⠠⠁⠏⠏⠇⠊⠉⠁⠞⠊⠕⠝ ⠠⠠⠊⠎ ⠠⠠⠍⠁⠙⠑ ⠠⠠⠁⠧⠁⠊⠇⠁⠃⠇⠑ ⠠⠠⠕⠝ ⠠⠠⠞⠓⠑ ⠠⠠⠁⠏⠏⠇⠊⠉⠁⠝⠞⠎
⠠⠠⠁⠉⠉⠕⠥⠝⠞ ⠠⠠⠁⠝⠙ ⠠⠠⠊⠝ ⠠⠠⠞⠓⠑ ⠠⠠⠁⠃⠎⠑⠝⠉⠑ ⠠⠠⠕⠋ ⠠⠠⠁ ⠠⠠⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑
⠠⠠⠎⠥⠍⠍⠁⠗⠽ ⠠⠠⠁⠎⠎⠥⠍⠑⠎ ⠠⠠⠞⠓⠁⠞ ⠠⠠⠞⠓⠑ ⠠⠠⠁⠏⠏⠇⠊⠉⠁⠝⠞ ⠠⠠⠓⠁⠎
⠠⠠⠉⠕⠍⠏⠇⠊⠑⠙ ⠠⠠⠺⠊⠞⠓ ⠠⠠⠁⠇⠇ ⠠⠠⠞⠓⠑ ⠠⠠⠗⠑⠗⠥⠊⠗⠑⠍⠑⠝⠞⠎ ⠠⠠⠕⠋ ⠠⠠⠞⠓⠊⠎
⠠⠠⠎⠑⠉⠞⠊⠕⠝⠲

⠠⠠⠠⠃⠊⠃⠇⠊⠕⠛⠗⠁⠋⠊⠁ ⠠⠠⠠⠃⠊⠃⠇⠊⠕⠛⠗⠁⠋⠊⠁ ⠠⠠⠠⠃⠊⠃⠇⠊⠕⠛⠗⠁⠋⠊⠁

⠠⠞⠓⠑ ⠠⠃⠇⠁⠉⠅ ⠙⠑⠁⠞⠓ ⠞⠗⠁⠏ ⠺⠁⠎ ⠁ ⠎⠞⠁⠞⠑ ⠞⠓⠁⠞ ⠞⠗⠁⠉⠑⠙ ⠞⠓⠑
⠺⠁⠽ ⠃⠁⠉⠅ ⠞⠕ ⠓⠊⠎ ⠓⠕⠍⠑⠞⠕⠺⠝ ⠁⠋⠞⠑⠗ ⠓⠑ ⠓⠁⠙ ⠙⠊⠑⠙ ⠁⠝⠙ ⠺⠁⠎
⠞⠕ ⠞⠕ ⠓⠊⠎ ⠋⠁⠞⠓⠑⠗ ⠲ ⠠⠞⠓⠑ ⠉⠗⠊⠍⠑ ⠙ ⠁ ⠉⠕⠍⠏⠁⠗⠁⠞⠊⠧⠑⠇⠽ ⠃⠇⠁⠉⠅ ⠠
⠺⠁⠎⠇⠑⠎⠎ ⠁ ⠉⠗⠊⠍⠑ ⠠⠃⠽ ⠞⠓⠁⠞ ⠞⠊⠍⠑ ⠠⠞⠓⠑ ⠉⠗⠊⠍⠑
⠠⠃⠇⠁⠉⠅⠎ ⠉⠗⠑⠁⠞⠑ ⠠⠞⠓⠑ ⠺⠁⠎⠇⠑⠎⠎⠝⠑⠎⠎ ⠃⠇⠁⠉⠅
⠠⠁ ⠃⠁⠙ ⠉⠓⠁⠗⠁⠉⠞⠑⠗ ⠁⠝⠙ ⠁ ⠙⠑⠞⠑⠎⠞⠁⠃⠇⠑ ⠉⠗⠊⠍⠑
⠠⠞⠓⠑ ⠠⠃⠇⠁⠉⠅ ⠠⠃⠁⠝⠙

⠠⠞⠓⠑ ⠠⠉⠕⠗⠑ ⠠⠁⠗⠉⠓⠊⠧⠑ ⠠⠕⠋ ⠠⠍⠁⠗⠊⠞⠊⠍⠑ ⠠⠞⠗⠁⠉⠑⠎ ⠠⠁⠝⠙
⠠⠞⠓⠑ ⠠⠍⠁⠗⠊⠞⠊⠍⠑ ⠠⠞⠊⠍⠑ ⠠⠎⠑⠗⠊⠑⠎ ⠠⠁⠗⠑ ⠠⠁ ⠠⠉⠕⠍⠍⠥⠝⠊⠞⠽
⠠⠕⠋ ⠠⠃⠕⠞⠓ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠍⠁⠗⠊⠞⠊⠍⠑ ⠠⠞⠊⠍⠑ ⠠⠎⠑⠗⠊⠑⠎ ⠠⠊⠎ ⠠⠁
⠠⠉⠕⠍⠍⠥⠝⠊⠞⠽ ⠠⠕⠋ ⠠⠍⠁⠗⠊⠞⠊⠍⠑ ⠠⠎⠉⠊⠑⠝⠞⠊⠎⠞⠎ ⠠⠁⠝⠙ ⠠⠑⠙⠥⠉⠁⠞⠕⠗⠎
⠠⠺⠓⠕ ⠠⠁⠗⠑ ⠠⠙⠑⠙⠊⠉⠁⠞⠑⠙ ⠠⠞⠕ ⠠⠞⠓⠑ ⠠⠎⠞⠥⠙⠽ ⠠⠕⠋ ⠠⠍⠁⠗⠊⠞⠊⠍⠑
⠠⠞⠊⠍⠑ ⠠⠎⠑⠗⠊⠑⠎ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠍⠁⠗⠊⠞⠊⠍⠑ ⠠⠞⠊⠍⠑ ⠠⠎⠑⠗⠊⠑⠎
⠠⠊⠝ ⠠⠞⠓⠑ ⠠⠉⠕⠝⠞⠑⠭⠞ ⠠⠕⠋ ⠠⠍⠁⠗⠊⠞⠊⠍⠑ ⠠⠎⠉⠊⠑⠝⠞⠊⠎⠞⠎ ⠠⠁⠝⠙
⠠⠍⠁⠗⠊⠞⠊⠍⠑ ⠠⠎⠉⠊⠑⠝⠞⠊⠎⠞⠎ ⠠⠁⠗⠑ ⠠⠁⠇⠎⠕ ⠠⠊⠝⠧⠕⠇⠧⠑⠙ ⠠⠊⠝
⠠⠞⠓⠑ ⠠⠙⠑⠧⠑⠇⠕⠏⠍⠑⠝⠞ ⠠⠕⠋ ⠠⠞⠑⠉⠓⠝⠊⠉⠁⠇ ⠠⠍⠑⠞⠓⠕⠙⠎ ⠠⠁⠝⠙
⠠⠞⠑⠉⠓⠝⠊⠉⠁⠇ ⠠⠎⠞⠁⠝⠙⠁⠗⠙⠎ ⠠⠞⠓⠁⠞ ⠠⠁⠗⠑ ⠠⠥⠎⠑⠙ ⠠⠊⠝ ⠠⠞⠓⠑

⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑, ⠃⠽ ⠙⠥⠍⠁⠎ ⠙⠑⠉⠇⠁⠗⠁⠞⠊⠕⠝, ⠞⠓⠑⠽ ⠓⠁⠧⠑
⠙⠑⠉⠇⠁⠗⠑⠙ ⠞⠓⠁⠞ ⠁ ⠞⠗⠁⠙⠑⠍⠁⠗⠅ ⠉⠕⠥⠇⠙ ⠝⠕⠞
⠒⠒ ⠞⠓⠑⠗⠑⠋⠕⠗⠑ ⠉⠓⠁⠝⠛⠑ ⠞⠓⠑ ⠉⠕⠝⠙⠊⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠊⠎ ⠒⠒

⠁. ⠙⠥⠍⠁⠎, ⠋⠊⠝⠁⠝⠉⠊⠁⠇ ⠎⠞⠁⠞⠑⠍⠑⠝⠞ ⠊⠎⠎⠥⠑⠙ ⠃⠽ ⠞⠓⠑ ⠉⠕⠍⠍⠊⠎⠎⠊⠕⠝
⠕⠋ ⠏⠥⠃⠇⠊⠉ ⠁⠉⠉⠕⠥⠝⠞⠎ ⠕⠋ ⠀⠀ ⠀ ⠀⠀⠀⠀⠀⠀ ⠀⠀⠀⠀⠀⠀

⠠⠠⠠

[illegible]

[illegible]

[illegible]

[illegible]

[

⠠⠓⠑⠗⠑ ⠊⠎ ⠁ ⠞⠁⠃⠇⠑ ⠉⠕⠍⠏⠁⠗⠊⠝⠛ ⠞⠓⠑ ⠏⠗⠑⠉⠊⠏⠊⠞⠁⠞⠊⠕⠝ ⠕⠋ ⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠙⠊⠑⠞⠎⠲

⠠⠞⠗⠑⠁⠞⠍⠑⠝⠞	⠠⠉⠁⠇⠕⠗⠊⠑⠎	⠠⠏⠗⠕⠞⠑⠊⠝	⠠⠉⠁⠗⠃⠕⠓⠽⠙⠗⠁⠞⠑⠎	⠠⠋⠁⠞
⠠⠙⠊⠑⠞ ⠁	⠼⠃⠚⠚	⠼⠁⠲⠼⠊ ⠛	⠼⠁⠲⠼⠊ ⠛	⠼⠁⠲⠼⠊ ⠛
⠠⠙⠊⠑⠞ ⠃	⠼⠑⠚⠚	⠼⠃⠲⠼⠑ ⠛	⠼⠃⠲⠼⠑ ⠛	⠼⠃⠲⠼⠑ ⠛
⠠⠙⠊⠑⠞ ⠉	⠼⠊⠚⠚	⠼⠉⠲⠼⠊ ⠛	⠼⠉⠲⠼⠊ ⠛	⠼⠉⠲⠼⠊ ⠛
⠠⠙⠊⠑⠞ ⠙	⠼⠁⠚⠚⠚	⠼⠙⠲⠼⠊ ⠛	⠼⠙⠲⠼⠊ ⠛	⠼⠙⠲⠼⠊ ⠛
⠠⠙⠊⠑⠞ ⠑	⠼⠁⠃⠚⠚	⠼⠑⠲⠼⠊ ⠛	⠼⠑⠲⠼⠊ ⠛	⠼⠑⠲⠼⠊ ⠛
⠠⠙⠊⠑⠞ ⠋	⠼⠁⠙⠚⠚	⠼⠋⠲⠼⠊ ⠛	⠼⠋⠲⠼⠊ ⠛	⠼⠋⠲⠼⠊ ⠛
⠠⠙⠊⠑⠞ ⠛	⠼⠁⠑⠚⠚	⠼⠛⠲⠼⠊ ⠛	⠼⠛⠲⠼⠊ ⠛	⠼⠛⠲⠼⠊ ⠛
⠠⠙⠊⠑⠞ ⠓	⠼⠁⠋⠚⠚	⠼⠓⠲⠼⠊ ⠛	⠼⠓⠲⠼⠊ ⠛	⠼⠓⠲⠼⠊ ⠛
⠠⠙⠊⠑⠞ ⠊	⠼⠁⠛⠚⠚	⠼⠊⠲⠼⠊ ⠛	⠼⠊⠲⠼⠊ ⠛	⠼⠊⠲⠼⠊ ⠛
⠠⠙⠊⠑⠞ ⠚	⠼⠁⠓⠚⠚	⠼⠚⠲⠼⠊ ⠛	⠼⠚⠲⠼⠊ ⠛	⠼⠚⠲⠼⠊ ⠛

·····: · ····· ····· · ··· ·····
· · ··· ····· ····· ·····. ··· ····
·· ··· ····· ····· ·· ·· · ·· ·····
····· ··· ··· ····· ··· ··· · ·····
··· ····· ····· · · ····· · ·····
·· ····· ····· ····· ·· ····· ·
· ·· ····· ····· ·· ····· ··· ·····
·····

·····. ····· ·· ····· · ·· ····· ··
· ····· ·····. ····· ·· ····· ·····
····· ····· ·· ····· ····· ·····
····· ····· · ····· ····· ·· ·····
····· ····· ····· ····· ····· ·
····· ····· ····· ····· ····· ·····
····· ····· ····· ····· ····· ·····
····· ····· ····· ····· ····· ·····

⠠⠞⠓⠑ ⠙⠁⠽ ⠕⠋ ⠠⠞⠓⠑⠕⠙⠕⠗⠑ ⠠⠍⠁⠝ ⠺⠁⠎ ⠞⠕ ⠠⠃⠑ ⠠⠁⠝ ⠑⠧⠑⠝⠞ ⠞⠓⠁⠞ ⠺⠕⠥⠇⠙ ⠞⠁⠅⠑ ⠏⠇⠁⠉⠑ ⠕⠝ ⠞⠓⠑ ⠎⠑⠧⠑⠝⠞⠓ ⠙⠁⠽ ⠕⠋ ⠞⠓⠑ ⠎⠑⠧⠑⠝⠞⠓ ⠍⠕⠝⠞⠓ ⠕⠋ ⠞⠓⠑ ⠁⠝⠝⠥⠁⠇ ⠉⠽⠉⠇⠑ ⠕⠋ ⠞⠓⠑ ⠎⠥⠝ ⠁⠎ ⠞⠓

⠠⠓⠊⠎ ⠑⠽⠑⠎ ⠺⠑⠗⠑ ⠎⠞⠊⠇⠇ ⠕⠏⠑⠝ ⠃⠥⠞ ⠎⠕⠍⠑⠺⠓⠁⠞ ⠉⠇⠕⠎⠑⠙ ⠺⠓⠑⠝ ⠞⠓⠑
⠽ ⠺⠑⠗⠑ ⠞⠁⠅⠑⠝ ⠁⠺⠁⠽ ⠃⠽ ⠎⠕⠍⠑ ⠞⠗⠁⠝⠎⠏⠕⠗⠞ ⠁⠛⠑⠝⠞. ⠠⠓⠑⠗⠑ ⠃⠥⠞
⠞⠓⠑ ⠞⠗⠁⠝⠎⠏⠕⠗⠞⠁⠞⠊⠕⠝ ⠺⠁⠎ ⠝⠕⠞ ⠙⠕⠝⠑ ⠁⠞ ⠁⠇⠇ ⠃⠽ ⠞⠓⠑ ⠞⠊⠍⠑ ⠠⠊ ⠺⠁⠎
⠃⠁⠉⠅ ⠁⠞ ⠞⠓⠑ ⠎⠞⠁⠞⠊⠕⠝ ⠁⠝⠙ ⠎⠁⠺ ⠞⠓⠑ ⠞⠗⠁⠊⠝ ⠙⠑⠏⠁⠗⠞ ⠺⠊⠞⠓ ⠁ ⠎⠞⠁⠝⠙⠁⠗⠙
⠎⠏⠁⠝⠊⠑⠇ ⠁⠝⠙ ⠞⠓⠑ ⠞⠗⠁⠊⠝ ⠺⠁⠎ ⠛⠕⠝⠑. ⠠⠊ ⠙⠕⠝⠞ ⠅⠝⠕⠺ ⠺⠓⠁⠞ ⠞⠕ ⠞⠓⠊⠝⠅
⠕⠋ ⠞⠓⠁⠞ ⠎⠞⠁⠞⠊⠕⠝ ⠁⠝⠙ ⠎⠁⠺ ⠞⠓⠑ ⠞⠗⠁⠊⠝ ⠙⠑⠏⠁⠗⠞ ⠁⠛⠁⠊⠝ ⠁⠞ ⠞⠓⠁⠞ ⠞⠊⠍⠑.
⠠⠓⠑⠗⠑ ⠺⠁⠎ ⠝⠕ ⠎⠞⠁⠞⠊⠕⠝ ⠁⠞ ⠁⠇⠇ ⠁⠝⠙ ⠞⠓⠑ ⠞⠗⠁⠊⠝ ⠙⠑⠏⠁⠗⠞⠑⠙ ⠁⠛⠁⠊⠝ ⠁⠝⠙
⠎⠁⠺ ⠞⠓⠑ ⠞⠗⠁⠊⠝ ⠙⠑⠏⠁⠗⠞ ⠁⠛⠁⠊⠝ ⠁⠞ ⠞⠓⠁⠞ ⠞⠊⠍⠑ ⠠⠊ ⠺⠁⠎ ⠝⠕⠞ ⠙⠕⠝⠑ ⠁⠞ ⠁⠇⠇
⠁⠝⠙ ⠞⠓⠑ ⠞⠗⠁⠊⠝ ⠙⠑⠏⠁⠗⠞⠑⠙

⠠ ⠓⠽⠏⠕⠞⠓⠑⠎⠊⠎ ⠊⠎ ⠞⠓⠑ ⠎⠑⠇⠑⠉⠞⠊⠕⠝ ⠕⠋ ⠁ ⠎⠏⠑⠉⠊⠋⠊⠉ ⠓⠽⠏⠕⠞⠓⠑⠎⠊⠎ ⠋⠗⠕⠍ ⠁ ⠏⠗⠑⠙⠑⠞⠑⠗⠍⠊⠝⠑⠙ ⠎⠑⠞ ⠕⠋ ⠁⠇⠞⠑⠗⠝⠁⠞⠊⠧⠑ ⠓⠽⠏⠕⠞⠓⠑⠎⠑⠎ ⠞⠓⠁⠞ ⠁⠗⠑ ⠉⠕⠝⠎⠊⠎⠞⠑⠝⠞ ⠺⠊⠞⠓ ⠁⠇⠇ ⠕⠃⠎⠑⠗⠧⠑⠙ ⠙⠁⠞⠁ ⠠⠓⠽⠏⠕⠞⠓⠑⠎⠊⠎ ⠞⠑⠎⠞⠊⠝⠛ ⠊⠎ ⠁ ⠉⠕⠍⠍⠕⠝ ⠏⠗⠁⠉⠞⠊⠉⠑ ⠊⠝ ⠎⠉⠊⠑⠝⠞⠊⠋⠊⠉ ⠗⠑⠎⠑⠁⠗⠉⠓ ⠁⠝⠙ ⠊⠞ ⠊⠎ ⠁ ⠉⠗⠊⠞⠊⠉⠁⠇ ⠎⠞⠑⠏ ⠊⠝ ⠞⠓⠑ ⠙⠑⠧⠑⠇⠕⠏⠍⠑⠝⠞ ⠕⠋ ⠁ ⠝⠑⠺ ⠙⠗⠥⠛ ⠞⠕ ⠙⠑⠞⠑⠗⠍⠊⠝⠑ ⠺⠓⠑⠞⠓⠑⠗ ⠁ ⠉⠁⠝⠙⠊⠙⠁⠞⠑ ⠙⠗⠥⠛ ⠊⠎ ⠎⠁⠋⠑ ⠁⠝⠙ ⠑⠋⠋⠑⠉⠞⠊⠧⠑ ⠋⠕⠗ ⠁ ⠛⠊⠧⠑⠝ ⠏⠁⠞⠊⠑⠝⠞ ⠁⠝⠙ ⠙⠊⠎⠑⠁⠎⠑ ⠁⠝⠙ ⠊⠞ ⠊⠎ ⠁⠇⠎⠕

⠠⠠⠠
⠠⠠⠠

⠉⠗⠑ ⠞⠓ ⠞⠇⠊⠍⠃⠇⠑ ⠃⠇⠥ ⠏⠗⠕⠞⠑⠉⠞⠊⠕⠝ ⠙⠥ ⠇⠁⠍⠃⠑⠁⠥ ⠞⠓ ⠊⠃
⠊⠉⠁ ⠞⠓⠁⠞ ⠞⠓⠑⠽ ⠉⠑⠝⠇⠊⠝⠑ ⠞⠕ ⠞⠗⠊⠃⠥⠞⠑ ⠁ ⠏⠁⠗⠁⠙⠊
⠉⠁ ⠉ ⠺⠊⠞⠓⠊⠝ ⠞⠓ ⠏⠗⠕⠞⠑⠉⠞⠊⠕⠝ ⠉⠕⠍⠏⠇⠑⠭ ⠃⠇⠥ ⠎⠑ ⠎⠕⠍⠑
⠥ ⠁ ⠉⠕⠍⠏⠁⠎⠎⠊⠕⠝⠁⠇ ⠞⠁ ⠺⠊⠇⠇. ⠞⠓⠑⠽⠗⠑ ⠏⠇⠁⠝⠞⠎ ⠞⠓⠁
⠎ ⠞⠓ ⠃⠇⠥ ⠉⠕⠍⠏⠇⠑⠞⠑ ⠙⠑ ⠝⠊⠞⠗⠕⠛⠑⠝ ⠞⠓⠁⠞ ⠞⠓⠑⠽ ⠉⠁
⠃⠁⠉⠞⠑⠗⠊⠁ ⠉⠕⠍⠏⠁⠗⠑⠙ ⠏⠗⠕⠙⠥⠉⠑ ⠞⠓⠑⠊⠗ ⠕⠺⠝ ⠎⠥⠛⠁⠗⠎ ⠁⠝⠙
⠞⠓⠑⠽ ⠁⠗⠑ ⠑⠭⠉⠗⠑⠞⠑⠙ ⠃⠽ ⠞⠓⠑ ⠏⠇⠁⠝⠞ ⠞⠓⠁⠞ ⠏⠗⠕⠙⠥⠉⠑ ⠊⠞
⠞⠕ ⠞⠓⠑ ⠁⠞⠍⠕⠎⠏⠓⠑⠗⠑ ⠺⠓⠑⠝ ⠞⠓⠑ ⠁⠊⠗ ⠊⠎ ⠙⠗⠽ ⠁⠝⠙ ⠞⠓⠑
⠏⠇⠁⠝⠞⠎ ⠁⠗⠑ ⠞⠁⠅⠑⠝ ⠕⠥⠞ ⠕⠋ ⠞⠓⠑ ⠎⠕⠊⠇ ⠁⠝⠙ ⠞⠓⠑ ⠺⠁⠞⠑⠗
⠞⠓⠑⠽ ⠁⠗⠑ ⠏⠇⠁⠉⠑⠙ ⠊⠝ ⠞⠓⠑ ⠎⠕⠊⠇ ⠁⠝⠙ ⠺⠁⠞⠑⠗ ⠞⠓⠁⠞ ⠁⠗⠑
⠝⠕⠞ ⠁⠇⠗⠑⠁⠙⠽ ⠏⠗⠑⠎⠑⠝⠞ ⠊⠝ ⠞⠓⠑ ⠁⠊⠗ ⠁⠝⠙ ⠞⠓⠑ ⠏⠇⠁⠝⠞⠎
⠞⠓⠁⠞ ⠁⠗⠑ ⠝⠕⠞ ⠁⠇⠗⠑⠁⠙⠽ ⠏⠗⠑⠎⠑⠝⠞ ⠊⠝ ⠞⠓⠑ ⠎⠕⠊⠇ ⠁⠝⠙ ⠺⠁⠞⠑⠗
⠞⠓⠑⠽ ⠁⠗⠑ ⠏⠇⠁⠉⠑⠙ ⠊⠝ ⠞⠓⠑ ⠎⠕⠊⠇ ⠁⠝⠙ ⠺⠁⠞⠑⠗ ⠞⠓⠁⠞
⠃⠑⠉⠁⠥⠎⠑ ⠞⠓⠑⠽ ⠏⠗⠕⠙⠥⠉⠑ ⠞⠓⠑⠊⠗ ⠕⠺⠝ ⠎⠥⠛⠁⠗⠎ ⠁⠝⠙ ⠺⠁⠞⠑⠗

⠠⠞⠓⠑ ⠠⠍⠑⠍⠕⠗⠽ ⠠⠏⠁⠇⠁⠉⠑ ⠠⠞⠕⠕⠅ ⠠⠞⠓⠗⠑⠑ ⠠⠽⠑⠁⠗⠎ ⠠⠞⠕ ⠠⠃⠥⠊⠇⠙ ⠠⠁ ⠠⠏⠁⠇⠁⠉⠑ ⠠⠕⠋ ⠠⠍⠑⠍⠕⠗⠽ ⠠⠁⠎ ⠠⠁ ⠠⠏⠇⠁⠉⠑ ⠠⠋⠕⠗

⠠⠊⠝⠞⠗⠕⠙⠥⠉⠞⠊⠕⠝ ⠞⠕ ⠠⠏⠗⠊⠝⠉⠊⠏⠇⠑⠎ ⠁⠝⠙ ⠠⠁⠏⠏⠇⠊⠉⠁⠞⠊⠕⠝⠎ ⠕⠋ ⠠⠃⠁⠉⠅⠏⠗⠕⠏⠁⠛⠁⠞⠊⠕⠝
⠞⠓⠑⠕⠗⠽ ⠺⠊⠞⠓ ⠠⠁⠏⠏⠇⠊⠉⠁⠞⠊⠕⠝⠎ ⠞⠕ ⠠⠎⠊⠍⠏⠇⠑ ⠠⠝⠑⠥⠗⠁⠇ ⠠⠝⠑⠞⠺⠕⠗⠅⠎ ⠁⠝⠙ ⠠⠙⠑⠑⠏
⠇⠑⠁⠗⠝⠊⠝⠛ ⠁⠇⠛⠕⠗⠊⠞⠓⠍⠎ ⠁⠝⠙ ⠞⠓⠑⠊⠗ ⠁⠏⠏⠇⠊⠉⠁⠞⠊⠕⠝⠎ ⠞⠕ ⠠⠉⠕⠍⠏⠥⠞⠑⠗ ⠠⠧⠊⠎⠊⠕⠝
⠠⠏⠗⠕⠃⠇⠑⠍⠎⠂ ⠠⠁⠎ ⠠⠁⠗⠞⠊⠋⠊⠉⠊⠁⠇ ⠠⠊⠝⠞⠑⠇⠇⠊⠛⠑⠝⠉⠑ ⠠⠗⠑⠎⠑⠁⠗⠉⠓ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝
⠠⠏⠗⠕⠛⠗⠑⠎⠎⠑⠎ ⠊⠝ ⠠⠍⠁⠉⠓⠊⠝⠑ ⠠⠇⠑⠁⠗⠝⠊⠝⠛ ⠁⠝⠙ ⠠⠙⠑⠑⠏ ⠠⠇⠑⠁⠗⠝⠊⠝⠛⠂ ⠠⠊⠞
⠠⠊⠎ ⠠⠁ ⠠⠎⠥⠃⠎⠑⠞ ⠕⠋ ⠠⠁⠗⠞⠊⠋⠊⠉⠊⠁⠇ ⠠⠊⠝⠞⠑⠇⠇⠊⠛⠑⠝⠉⠑⠂ ⠠⠃⠥⠞ ⠠⠊⠞ ⠠⠙⠊⠋⠋⠑⠗⠎
⠠⠋⠗⠕⠍ ⠠⠞⠓⠑ ⠠⠍⠕⠗⠑ ⠠⠞⠗⠁⠙⠊⠞⠊⠕⠝⠁⠇ ⠠⠁⠏⠏⠗⠕⠁⠉⠓⠑⠎ ⠠⠞⠕ ⠠⠍⠁⠉⠓⠊⠝⠑
⠠⠇⠑⠁⠗⠝⠊⠝⠛ ⠠⠊⠝ ⠠⠞⠓⠁⠞ ⠠⠊⠞ ⠠⠁⠇⠇⠕⠺⠎ ⠠⠞⠓⠑ ⠠⠍⠁⠉⠓⠊⠝⠑ ⠠⠞⠕
⠠⠏⠗⠕⠛⠗⠁⠍⠍⠑⠗ ⠠⠊⠞⠎⠑⠇⠋⠂ ⠠⠁⠝⠙ ⠠⠊⠝ ⠠⠞⠓⠊⠎ ⠠⠎⠑⠝⠎⠑⠂ ⠠⠊⠞ ⠠⠉⠁⠝ ⠠⠃⠑
⠠⠉⠕⠝⠎⠊⠙⠑⠗⠑⠙ ⠠⠁ ⠠⠋⠕⠗⠍ ⠕⠋ ⠠⠁⠗⠞⠊⠋⠊⠉⠊⠁⠇ ⠠⠊⠝⠞⠑⠇⠇⠊⠛⠑⠝⠉⠑⠲

⠠⠁ ⠠⠏⠗⠕⠛⠗⠁⠍ ⠠⠞⠓⠁⠞ ⠠⠉⠁⠝ ⠠⠇⠑⠁⠗⠝ ⠠⠋⠗⠕⠍ ⠠⠑⠭⠏⠑⠗⠊⠑⠝⠉⠑ ⠠⠞⠕⠲
⠠⠠⠠⠁ ⠠⠏⠗⠕⠛⠗⠁⠍ ⠠⠊⠎ ⠠⠁ ⠠⠎⠑⠞ ⠕⠋ ⠠⠊⠝⠎⠞⠗⠥⠉⠞⠊⠕⠝⠎ ⠠⠞⠓⠁⠞ ⠠⠍⠁⠅⠑⠎ ⠠⠁
⠠⠉⠕⠍⠏⠥⠞⠑⠗ ⠠⠙⠕ ⠠⠁ ⠠⠉⠑⠗⠞⠁⠊⠝ ⠠⠞⠁⠎⠅⠂ ⠠⠁⠝⠙ ⠠⠍⠁⠅⠑⠎ ⠠⠁ ⠠⠉⠕⠍⠏⠥⠞⠑⠗
⠠⠙⠕ ⠠⠁ ⠠⠙⠊⠋⠋⠑⠗⠑⠝⠞ ⠠⠞⠁⠎⠅⠲ ⠠⠠⠠⠁ ⠠⠏⠗⠕⠛⠗⠁⠍ ⠠⠍⠁⠽ ⠠⠃⠑ ⠠⠁⠇⠎⠕ ⠠⠃⠑
⠠⠉⠕⠝⠎⠊⠙⠑⠗⠑⠙ ⠠⠁ ⠠⠙⠁⠞⠁ ⠠⠞⠽⠏⠑⠲

A photograph of a large, ancient stone structure, likely a temple or monumental building, with a prominent triangular pediment and multiple columns. The structure is set against a backdrop of trees and a clear sky. In the foreground, there is a paved area with a few scattered stones. The overall scene suggests a historical or archaeological site.

⠠⠁⠞ ⠺⠓⠁⠞ ⠏⠕⠊⠝⠞ ⠙⠕⠑⠎ ⠞⠓⠑ ⠏⠗⠕⠃⠇⠑⠍ ⠎⠑⠞ ⠕⠥⠞ ⠞⠕ ⠎⠕⠇⠧⠑ ⠞⠓⠑ ⠏⠗⠕⠃⠇⠑⠍ ⠁⠎

⠠⠓ ⠞⠗⠁⠙⠑ ⠉⠕⠍⠍⠊⠎⠎⠊⠕⠝ ⠞⠕ ⠗⠑⠧⠊⠑⠺ ⠁⠝ ⠁⠍⠑⠝⠙⠍⠑⠝⠞ ⠞⠕
⠠⠠⠠

⠠⠠⠊⠝⠞⠑⠛⠗⠁⠉⠊⠕⠝ ⠙⠑ ⠇⠁ ⠠⠠⠉⠁⠝⠞⠊⠙⠁⠙ ⠙⠑ ⠍⠁⠞⠗⠊⠉⠥⠇⠁⠎ ⠙⠑ ⠇⠁ ⠠⠠⠊⠝⠙⠥⠎⠞⠗⠊⠁ ⠙⠑ ⠇⠁ ⠠⠠⠁⠇⠊⠍⠑⠝⠞⠁⠉⠊⠕⠝ ⠙⠑ ⠇⠁ ⠠⠠⠁⠛⠗⠕⠏⠑⠉⠥⠁⠗⠊⠁ ⠙⠑ ⠇⠁ ⠠⠠⠁⠗⠛⠑⠝⠞⠊⠝⠁ ⠠⠠⠁⠗⠞⠑⠎⠁⠝⠁⠇

⠠⠁ ⠠⠠⠊⠠⠠⠍⠠⠁⠠⠁ ⠠⠁ ⠠⠠⠠⠁⠠ ⠠⠠⠁ ⠠⠊⠠⠠ ⠠⠁ ⠠⠁⠠⠠⠠⠁⠠⠊⠠⠁ ⠠⠁ ⠠⠠⠁
⠠⠠⠠ ⠠⠠⠊⠠⠠⠁⠠⠠⠠⠁⠠⠠ ⠠⠠⠁ ⠠⠠⠊⠠⠠ ⠠⠠⠠⠁⠠⠠⠠⠁⠠⠠ ⠠⠠⠁ ⠠⠠⠊⠠⠠
⠠⠠⠁ ⠠⠠ ⠠⠠⠁⠠⠠⠁⠠⠠. ⠠⠠ ⠠ ⠠⠁ ⠠⠠⠁⠠⠠⠁⠠⠠⠁⠠⠠ ⠠⠠⠊⠠⠁ ⠠⠠⠠⠁ ⠠⠠⠠
⠠⠠ ⠠⠠⠁ ⠠⠠⠊ ⠠⠠⠁⠠⠠⠁⠠⠠. ⠠⠠⠠⠁⠠⠠⠁⠠⠠⠁⠠⠠ ⠠⠠⠁⠠⠠⠁⠠⠠⠁⠠⠠ ⠠⠠⠁⠠⠠⠁⠠⠠⠁⠠⠠
⠠⠠⠁⠠⠠⠁⠠⠠⠁⠠⠠

⠠⠞⠓⠑ ⠠⠍⠁⠝ ⠺⠓⠕ ⠠⠅⠝⠑⠺ ⠠⠞⠕⠕ ⠍⠥⠉⠓ ⠀ ⠀ ⠀

⠠⠁⠗⠞⠊⠉⠇⠑ ⠊⠊⠊. ⠠⠊⠝ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠊⠊⠊, ⠠⠞⠓⠑ ⠠⠉⠕⠝⠎⠞⠊⠞⠥⠞⠊⠕⠝ ⠕⠋ ⠠⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁, ⠠⠺⠑ ⠠⠞⠓⠑ ⠠⠏⠑⠕⠏⠇⠑ ⠕⠋ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎, ⠠⠊⠝ ⠠⠕⠗⠙⠑⠗ ⠞⠕ ⠠⠋⠕⠗⠍ ⠁ ⠠⠍⠕⠗⠑ ⠠⠏⠻⠋⠑⠉⠞ ⠠⠥⠝⠊⠕⠝, ⠠⠁⠝⠙ ⠠⠞⠕ ⠠⠏⠗⠕⠍⠕⠞⠑ ⠠⠙⠕⠍⠑⠎⠞⠊⠉ ⠠⠞⠗⠁⠝⠗⠥⠇⠊⠞⠽, ⠠⠊⠝⠎⠥⠗⠑ ⠠⠙⠕⠍⠑⠎⠞⠊⠉ ⠠⠞⠗⠁⠝⠗⠥⠇⠊⠞⠽, ⠠⠁⠝⠙ ⠠⠞⠕ ⠠⠎⠑⠉⠥⠗⠑ ⠠⠞⠓⠑ ⠠⠃⠇⠑⠎⠎⠊⠝⠛⠎ ⠠⠕⠋ ⠠⠇⠊⠃⠑⠗⠞⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠉⠕⠝⠎⠑⠗⠧⠁⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠝⠁⠞⠥⠗⠁⠇ ⠠⠗⠑⠎⠕⠥⠗⠉⠑⠎ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠎⠑⠧⠑⠗⠁⠇ ⠠⠎⠞⠁⠞⠑⠎.

⠠⠎⠑⠉⠞⠊⠕⠝ ⠠⠊. ⠠⠁⠗⠞⠊⠉⠇⠑⠎ ⠠⠁⠝⠙ ⠠⠠⠁⠍⠑⠝⠙⠍⠑⠝⠞⠎.

⠠⠎⠑⠉⠞⠊⠕⠝ ⠠⠊⠊. ⠠⠁⠍⠑⠝⠙⠍⠑⠝⠞⠎ ⠠⠞⠕ ⠠⠞⠓⠑ ⠠⠉⠕⠝⠎⠞⠊⠞⠥⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠠⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁, ⠠⠺⠑ ⠠⠞⠓⠑ ⠠⠏⠑⠕⠏⠇⠑ ⠕⠋ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎, ⠠⠊⠝ ⠠⠕⠗⠙⠑⠗ ⠞⠕ ⠠⠋⠕⠗⠍ ⠁ ⠠⠍⠕⠗⠑ ⠠⠏⠻⠋⠑⠉⠞ ⠠⠥⠝⠊⠕⠝, ⠠⠁⠝⠙ ⠠⠞⠕ ⠠⠏⠗⠕⠍⠕⠞⠑ ⠠⠙⠕⠍⠑⠎⠞⠊⠉ ⠠⠞⠗⠁⠝⠗⠥⠇⠊⠞⠽, ⠠⠊⠝⠎⠥⠗⠑ ⠠⠙⠕⠍⠑⠎⠞⠊⠉ ⠠⠞⠗⠁⠝⠗⠥⠇⠊⠞⠽, ⠠⠁⠝⠙ ⠠⠞⠕ ⠠⠎⠑⠉⠥⠗⠑ ⠠⠞⠓⠑ ⠠⠃⠇⠑⠎⠎⠊⠝⠛⠎ ⠠⠕⠋ ⠠⠇⠊⠃⠑⠗⠞⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠉⠕⠝⠎⠑⠗⠧⠁⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠝⠁⠞⠥⠗⠁⠇ ⠠⠗⠑⠎⠕⠥⠗⠉⠑⠎ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠎⠑⠧⠑⠗⠁⠇ ⠠⠎⠞⠁⠞⠑⠎.

⠠⠎⠑⠉⠞⠊⠕⠝ ⠠⠊⠊⠊. ⠠⠁⠍⠑⠝⠙⠍⠑⠝⠞⠎ ⠠⠞⠕ ⠠⠞⠓⠑ ⠠⠉⠕⠝⠎⠞⠊⠞⠥⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠠⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁, ⠠⠺⠑ ⠠⠞⠓⠑ ⠠⠏⠑⠕⠏⠇⠑ ⠕⠋ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎, ⠠⠊⠝ ⠠⠕⠗⠙⠑⠗ ⠞⠕ ⠠⠋⠕⠗⠍ ⠁ ⠠⠍⠕⠗⠑ ⠠⠏⠻⠋⠑⠉⠞ ⠠⠥⠝⠊⠕⠝, ⠠⠁⠝⠙ ⠠⠞⠕ ⠠⠏⠗⠕⠍⠕⠞⠑ ⠠⠙⠕⠍⠑⠎⠞⠊⠉ ⠠⠞⠗⠁⠝⠗⠥⠇⠊⠞⠽, ⠠⠊⠝⠎⠥⠗⠑ ⠠⠙⠕⠍⠑⠎⠞⠊⠉ ⠠⠞⠗⠁⠝⠗⠥⠇⠊⠞⠽, ⠠⠁⠝⠙ ⠠⠞⠕ ⠠⠎⠑⠉⠥⠗⠑ ⠠⠞⠓⠑ ⠠⠃⠇⠑⠎⠎⠊⠝⠛⠎ ⠠⠕⠋ ⠠⠇⠊⠃⠑⠗⠞⠽ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠉⠕⠝⠎⠑⠗⠧⠁⠞⠊⠕⠝ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠝⠁⠞⠥⠗⠁⠇ ⠠⠗⠑⠎⠕⠥⠗⠉⠑⠎ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠎⠑⠧⠑⠗⠁⠇ ⠠⠎⠞⠁⠞⠑⠎.

· ··
·· ·· ··

⠠⠞⠓⠑ ⠙⠁⠗⠅ ⠺⠁⠎ ⠁ ⠇⠊⠛⠓⠞ ⠞⠓⠁⠞ ⠉⠁⠗⠗⠊⠑⠙ ⠞⠓⠑ ⠍⠁⠝ ⠞⠓⠗⠕⠥⠛⠓ ⠁ ⠞⠊⠍⠑ ⠞⠓⠁⠞ ⠍⠁⠙⠑ ⠁⠇⠇ ⠞⠓⠑ ⠙⠊⠋⠋⠑⠗⠑⠝⠉⠑ ⠊⠝ ⠞⠓⠑ ⠺⠕⠗⠇⠙ ⠁⠝⠙ ⠞⠓⠑ ⠙⠊⠋⠋⠑⠗⠑⠝⠉⠑ ⠊⠝ ⠞⠓⠑ ⠺⠕⠗⠇⠙ ⠺⠁⠎

⠠⠞⠓⠑ ⠠⠁⠗⠞ ⠕⠋ ⠠⠍⠑⠁⠎⠥⠗⠊⠝⠛ ⠠⠞⠊⠍⠑ ⠍⠁⠝

⠠⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠏⠎⠽⠉⠓⠊⠁⠞⠗⠊⠉ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠕⠋ ⠠⠉⠇⠊⠝⠊⠉⠁⠇ ⠠⠏⠎⠽⠉⠓⠕⠇⠕⠛⠽
⠨⠠⠁⠠⠏⠠⠠⠎ ⠕⠋ ⠠⠍⠑⠙⠊⠉⠊⠝⠑ ⠠⠁⠝⠙ ⠠⠎⠥⠗⠛⠑⠗⠽ ⠨⠠⠁⠠⠎⠠⠠ ⠠⠁⠝⠙ ⠠⠁⠠⠏⠠⠠⠎ ⠠⠁⠗⠑ ⠠⠁⠍⠕⠝⠛
⠠⠞⠓⠑ ⠠⠍⠕⠎⠞ ⠠⠉⠕⠍⠍⠕⠝⠇⠽ ⠠⠥⠎⠑⠙ ⠠⠞⠑⠎⠞⠎ ⠠⠊⠝ ⠠⠏⠎⠽⠉⠓⠊⠁⠞⠗⠽ ⠁⠗⠑ ⠠⠞⠓⠑
⠠⠎⠞⠁⠞⠊⠎⠞⠊⠉⠁⠇ ⠠⠍⠑⠞⠓⠕⠙⠎ ⠠⠞⠓⠁⠞ ⠠⠁⠗⠑ ⠠⠞⠕⠕ ⠠⠎⠏⠑⠉⠊⠁⠇⠊⠵⠑⠙ ⠠⠁⠝⠙ ⠠⠞⠕⠕
⠠⠉⠕⠍⠏⠇⠑⠭ ⠠⠋⠕⠗ ⠠⠞⠓⠑ ⠠⠁⠧⠑⠗⠁⠛⠑ ⠠⠏⠎⠽⠉⠓⠕⠇⠕⠛⠊⠎⠞ ⠠⠞⠕ ⠠⠥⠝⠙⠑⠗⠎⠞⠁⠝⠙
⠠⠁⠝⠙ ⠠⠞⠓⠑⠗⠁⠏⠊⠎⠞ ⠠⠞⠕ ⠠⠏⠗⠕⠧⠊⠙⠑ ⠠⠁ ⠠⠉⠕⠓⠑⠗⠑⠝⠞ ⠠⠞⠓⠑⠕⠗⠽
⠠⠕⠋ ⠠⠓⠥⠍⠁⠝ ⠠⠍⠕⠞⠊⠧⠁⠞⠊⠕⠝ ⠠⠁⠝⠙ ⠠⠃⠑⠓⠁⠧⠊⠕⠗
⠠⠠⠠

⠠⠠⠠
⠠⠠⠠

⠠⠞⠓ ⠠⠍⠁ ⠙ ⠠⠉⠁⠎⠁ ⠠⠍⠁⠝⠁ ⠠⠍⠁⠛⠊ ⠠⠞⠁ ⠠⠎⠊⠍⠏⠁⠝⠊ ⠠⠙⠑ ⠠⠏⠁⠗⠁
⠠⠍⠁⠝⠁ ⠠⠙⠑ ⠠⠏⠁⠗⠁ ⠠⠞⠁ ⠠⠏⠁⠗⠁⠝⠁ . ⠠ ⠠⠁⠍⠁⠗ ⠠⠁⠍⠁⠗⠁⠍
⠠⠍⠑ ⠠⠎ ⠠⠃⠁⠃⠁⠇ ⠠⠇⠑ ⠠⠠⠁⠇⠃⠁ ⠠⠠⠍⠁⠝⠁⠎⠇⠁ ⠠⠏⠁⠃ ⠠⠎⠊⠍
⠠⠞⠁ ⠙ ⠠⠍⠁⠝⠁⠎⠇⠁⠃⠁ ⠠⠞⠁⠛⠁⠗ ⠠⠎⠁ ⠠⠓ ⠙ ⠠⠊⠝⠙⠁⠛⠁⠞⠁
⠠⠞ ⠠⠉⠁⠃⠑⠠⠇ ⠙ ⠠⠠⠊⠝⠙⠊⠛⠁⠞⠁ ⠠⠠⠁⠗⠛⠁⠝⠞⠊⠝⠁ ⠠⠗⠑⠞⠁ ⠠⠎⠁
⠠⠃⠁⠇⠞⠁ ⠙ ⠠⠇⠁⠎ ⠠⠍⠁⠙⠗⠑⠎ ⠠⠃⠁⠎⠁⠙⠁⠎ ⠠⠑⠝ ⠠⠇⠁ ⠠⠠⠁⠝⠙⠑⠎
⠠⠞⠁ ⠠⠎⠑ ⠠⠏⠁⠎⠁⠃⠁ ⠠⠇⠁ ⠠⠠⠁⠗⠛⠑⠝⠞⠊⠝⠁ ⠠⠉⠁⠎⠁ ⠠⠏⠁⠗⠁
⠠⠞⠁⠝⠞⠁⠎ ⠠⠁⠝⠙⠁⠍⠁⠝ ⠠⠍⠁⠎⠁ ⠙ ⠠⠏⠁⠏⠁⠎ ⠠⠏⠁⠃ ⠠⠁ ⠠⠞⠁
⠠⠁ ⠠⠇⠁⠎ ⠠⠁ ⠠⠏⠁⠗⠁⠇⠁⠎ ⠠⠠⠁⠝⠙⠑⠎⠞⠁⠝⠞⠑⠎ ⠙ ⠠⠍⠁⠝⠁⠎⠇⠁⠃⠁
⠠⠍⠁ ⠠⠞⠁ ⠠⠏⠁⠗⠁⠙⠁⠎ ⠠⠉⠁⠎⠁ ⠠⠃⠁ ⠠⠍⠁⠝⠁⠎⠇⠁⠃⠁ ⠠⠁ ⠠⠁⠍⠁⠗
⠠⠁ ⠠⠏⠁⠃ ⠠⠍⠁⠝⠁ ⠠⠠⠁⠇⠃⠁ ⠠⠠⠍⠁⠝⠁⠎⠇⠁ ⠠⠠⠍⠁⠝⠁⠎⠇⠁ . ⠠⠁⠍⠁⠗
⠠⠁⠍⠁⠗⠁⠍ ⠠⠍⠁⠝⠁

⠠⠞⠓⠑⠗⠑⠋⠕⠗⠑⠂ ⠺⠓⠑⠝ ⠠⠊ ⠓⠁⠙ ⠙⠕⠝⠑ ⠁ ⠋⠥⠇⠇ ⠞⠗⠁⠝⠎⠉⠗⠊⠏⠞
⠕⠋ ⠠⠍⠁⠗⠽⠇⠁⠝⠙ ⠠⠎⠞⠁⠞⠥⠞⠑ ⠗⠑⠇⠁⠞⠊⠝⠛ ⠞⠕ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠗⠑⠧⠕⠇⠥⠞⠊⠕⠝
⠁⠝⠙ ⠕⠋ ⠞⠓⠑ ⠠⠉⠕⠝⠎⠞⠊⠞⠥⠞⠊⠕⠝ ⠠⠺⠓⠑⠝ ⠠⠊ ⠺⠁⠎ ⠁ ⠉⠓⠊⠇⠙⠂
⠠⠊ ⠓⠁⠙ ⠝⠑⠧⠑⠗ ⠓⠑⠁⠗⠙ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠗⠑⠧⠕⠇⠥⠞⠊⠕⠝⠁⠗⠽
⠠⠺⠁⠗⠂ ⠠⠊ ⠓⠁⠙ ⠝⠑⠧⠑⠗ ⠓⠑⠁⠗⠙ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠗⠑⠧⠕⠇⠥⠞⠊⠕⠝
⠁⠗⠽

⠠⠝ ⠠⠊⠊⠊⠓ ⠃⠠⠠⠞ ⠠⠎ ⠠⠎⠊⠠⠠⠃⠠⠃⠃⠠⠎ ⠠⠊⠊⠊⠓ ⠠⠠⠠⠎⠃⠠⠠⠞
⠠⠎ ⠠⠠⠃⠠⠃⠠⠠⠞⠠⠠⠎⠠⠃⠠⠃⠠⠎ ⠠⠠⠠⠞⠠⠊ ⠠⠠⠃⠠⠠⠞ ⠠⠝ ⠠

[illegible]

[illegible]

[illegible]

⠠⠃⠑ ⠠⠁ ⠠⠉⠕⠍⠏⠇⠑⠞⠑⠙ ⠠⠞⠕ ⠠⠇⠁⠞⠑⠭ ⠠⠊⠎ ⠠⠞⠓⠑ ⠠⠉⠕⠍⠏⠁⠝⠽⠄⠎ ⠠⠏⠗⠊⠍⠁⠗⠽
⠠⠎⠞⠁⠞⠑ ⠠⠍⠁⠉⠓⠊⠝⠑ ⠠⠁⠝⠙ ⠠⠞⠓⠑ ⠠⠇⠁⠞⠑⠭ ⠠⠁ ⠠⠉⠕⠍⠏⠁⠝⠽⠄⠎ ⠠⠍⠁⠉⠓⠊⠝⠑
⠠⠓⠁⠎ ⠠⠞⠓⠗⠑⠑ ⠠⠁⠉⠞⠊⠧⠑ ⠠⠎⠞⠁⠞⠑⠎⠲ ⠠⠠⠞⠓⠑ ⠠⠎⠞⠁⠞⠑ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠥⠝⠊⠕⠝⠠⠠ ⠠⠊⠎
⠠⠞⠓⠑ ⠠⠎⠞⠁⠞⠑ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠥⠝⠊⠕⠝⠲ ⠠⠠⠞⠓⠑ ⠠⠎⠞⠁⠞⠑ ⠠⠕⠋ ⠠⠞⠓⠑ ⠠⠥⠝⠊⠕⠝⠠⠠
⠠⠊⠎

⠠⠁⠇⠇ ⠗⠊⠛⠓⠞⠎ ⠗⠑⠎⠑⠗⠧⠑⠙⠲ ⠠⠎⠕⠍⠑ ⠗⠊⠛⠓⠞⠎ ⠍⠁⠽ ⠃⠑ ⠞⠗⠁⠝⠎⠋⠑⠗⠗⠑⠙ ⠕⠗ ⠞⠗⠁⠝⠎⠍⠊⠞⠞⠑⠙ ⠊⠝ ⠁⠝⠽ ⠋⠕⠗⠍ ⠕⠗ ⠃⠽ ⠁⠝⠽ ⠍⠑⠁⠝⠎ ⠞⠕ ⠅⠝⠕⠺⠝ ⠁⠋⠞⠑⠗ ⠞⠓⠊⠎ ⠙⠁⠞⠑ ⠊⠝⠉⠇⠥⠙⠊⠝⠛ ⠃⠥⠞ ⠝⠕⠞ ⠇⠊⠍⠊⠞⠑⠙ ⠞⠕ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠗⠞⠊⠎⠞⠎ ⠁⠝⠙ ⠠⠁⠥⠞⠓⠕⠗⠎ ⠠⠁⠝ ⠠⠁⠗⠞⠊⠎⠞ ⠎⠓⠁⠇⠇ ⠓⠁⠧⠑ ⠞⠓⠑ ⠗⠊⠛⠓⠞ ⠞⠕ ⠥⠎⠑ ⠁⠝⠙ ⠁⠥⠞⠓⠕⠗⠊⠵⠑

⠠⠠⠠
⠠⠠⠠

⠠ ⠠⠞⠓⠑ ⠎⠞⠁⠞⠑⠍⠑⠝⠞ ⠕⠋ ⠞⠓⠑ ⠠⠉⠕⠍⠍⠊⠎⠎⠊⠕⠝ ⠺⠁⠎ ⠞⠓⠁⠞ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠉⠕⠇⠕⠝⠊⠎⠞⠎ ⠙⠊⠙ ⠝⠕⠞ ⠓⠁⠧⠑ ⠁ ⠎⠞⠁⠝⠙⠁⠗⠙ ⠕⠋ ⠞⠓⠑⠊⠗ ⠕⠺⠝ ⠞⠕ ⠺⠓⠊⠉⠓ ⠞⠓⠑⠽ ⠁⠎⠏⠊⠗⠑⠙ ⠠⠠⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠉⠕⠇⠕⠝⠊⠎⠞⠎ ⠠⠠ ⠠⠁⠝⠙ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠉⠕⠇⠕⠝⠊⠎⠞⠎ ⠠⠠ ⠺⠑⠗⠑ ⠞⠓⠑ ⠠⠠⠞⠓⠊⠗⠞⠑⠑⠝ ⠠⠉⠕⠇⠕⠝⠊⠎⠞⠎ ⠠⠠ ⠁⠝⠙ ⠞⠓⠑ ⠠⠠⠎⠑⠧⠑⠝⠞⠑⠑⠝ ⠠⠉⠕⠇⠕⠝⠊⠎⠞⠎ ⠠⠠ ⠺⠓⠑⠝ ⠞⠓⠑⠽ ⠺⠑⠗⠑ ⠁⠎⠅⠑⠙ ⠺⠓⠁⠞ ⠞⠓⠑⠽ ⠺⠁⠝⠞⠑⠙ ⠞⠕ ⠙⠕ ⠁⠝⠙ ⠞⠓⠑⠽ ⠙⠑⠉⠇⠁⠗⠑⠙ ⠞⠓⠁⠞ ⠞⠓⠑⠽ ⠙⠊⠙ ⠝⠕⠞ ⠺⠁⠝⠞ ⠞⠕ ⠙⠕ ⠁⠝⠽⠞⠓⠊⠝⠛ ⠁⠞ ⠁⠇⠇ ⠞⠓⠁⠞ ⠺⠁⠎ ⠝⠕⠞ ⠞⠓⠑⠊⠗ ⠕⠺⠝ ⠺⠊⠎⠓ ⠁⠝⠙

⠠⠁ ⠉⠇⠁⠗⠊⠎⠎⠁ ⠞⠗⠊⠑⠙ ⠞⠕ ⠃⠕⠕⠅ ⠓⠊⠍ ⠊⠝ ⠎⠑⠧⠑⠗⠁⠇
⠠⠁ ⠇⠁⠞⠑ ⠍⠊⠙⠙⠇⠑⠃⠕⠗⠑ ⠏⠓⠁⠎⠑ ⠍⠁⠙⠑ ⠁ ⠉⠗⠁⠉⠅
⠉⠁⠍⠑ ⠁⠇⠕⠝⠛ ⠁⠝⠙ ⠎⠁⠊⠙ ⠠⠊ ⠺⠁⠎ ⠝⠕⠞ ⠎⠥⠗⠑
⠺⠓⠁⠞ ⠠⠊ ⠺⠁⠝⠞⠑⠙ ⠞⠕ ⠙⠕ ⠁⠝⠙ ⠠⠊ ⠙⠊⠙⠝⠞
⠠⠁ ⠍⠁⠝ ⠁ ⠇⠕⠝⠛ ⠞⠊⠍⠑ ⠁⠛⠕ ⠁⠝⠙ ⠠⠊ ⠙⠊⠙⠝⠞
⠅⠝⠕⠺ ⠺⠓⠁⠞ ⠠⠊ ⠺⠁⠝⠞⠑⠙ ⠞⠕ ⠙⠕ ⠃⠥⠞ ⠠⠊ ⠙⠊⠙⠝⠞
⠃⠑⠉⠁⠥⠎⠑ ⠠⠊ ⠙⠊⠙⠝⠞ ⠅⠝⠕⠺ ⠺⠓⠁⠞ ⠠⠊ ⠺⠁⠝⠞⠑⠙
⠞⠕ ⠙⠕ ⠁⠝⠙ ⠠⠊ ⠙⠊⠙⠝⠞ ⠅⠝⠕⠺ ⠺⠓⠁⠞ ⠠⠊ ⠺⠁⠝⠞⠑⠙
⠞⠕ ⠙⠕

⠠⠊⠍⠏⠁⠉⠞ ⠠⠋⠁⠉⠞⠕⠗⠎ ⠕⠝ ⠠⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑ ⠠⠁⠝⠁⠇⠽⠎⠊⠎ ⠕⠋ ⠠⠙⠗⠥⠛ ⠠⠥⠎⠑ ⠠⠊⠝ ⠠⠎⠕⠥⠞⠓ ⠠⠁⠋⠗⠊⠉⠁

⠠⠁⠃⠎⠞⠗⠁⠉⠞

⠠⠁ ⠠⠉⠕⠍⠏⠗⠑⠓⠑⠝⠎⠊⠧⠑ ⠠⠁⠝⠁⠇⠽⠎⠊⠎ ⠕⠋ ⠠⠙⠗⠥⠛ ⠠⠥⠎⠑ ⠠⠊⠝ ⠠⠎⠕⠥⠞⠓ ⠠⠁⠋⠗⠊⠉⠁ ⠠⠎⠓⠕⠥⠇⠙ ⠠⠃⠑

⠠⠉⠓⠁⠏⠞⠑⠗ ⠠⠠⠠

⠠⠞⠓⠑ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠗⠎ ⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇

⠠⠞⠓⠑ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠗⠎ ⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇ ⠠⠊⠝⠉⠕⠗⠏⠕⠗⠁⠞⠑⠙
⠁ ⠠⠏⠥⠃⠇⠊⠉ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠺⠊⠞⠓ ⠠⠁ ⠠⠗⠑⠉⠕⠗⠙ ⠕⠋ ⠠⠎⠥⠉⠉⠑⠎⠎⠋⠥⠇
⠠⠏⠗⠕⠙⠥⠉⠞⠊⠕⠝ ⠕⠋ ⠠⠞⠓⠑ ⠠⠋⠊⠗⠎⠞ ⠠⠁⠝⠝⠥⠁⠇ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠗⠎
⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇ ⠠⠉⠕⠝⠋⠑⠗⠑⠝⠉⠑ ⠕⠝ ⠠⠞⠓⠑ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠗⠎
⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠕⠋ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠗⠎
⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠕⠋ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠗⠎
⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠠⠊⠝⠉⠲ ⠠⠁ ⠠⠙⠑⠇⠁⠺⠁⠗⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠠⠍⠁⠝⠥⠋⠁⠉⠞⠥⠗⠑⠗⠎ ⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇ ⠠⠉⠕⠍⠏⠁⠝⠽
⠠⠁ ⠠⠙⠑⠇⠁⠺⠁⠗⠑ ⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠠⠁⠝⠙ ⠠⠁ ⠠⠏
⠠⠇⠥⠃⠇⠊⠉ ⠠⠉⠕⠍⠏⠁⠝⠽ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠋⠕⠥⠝⠙⠑⠙ ⠠⠕⠝ ⠠⠁⠏⠗⠊⠇ ⠼⠁⠁⠂ ⠼⠁⠛⠛⠼ ⠠⠃⠽
⠠⠞⠓⠑ ⠠⠎⠞⠁⠞⠑ ⠠⠕⠋ ⠠⠙⠑⠇⠁⠺⠁⠗⠑ ⠠⠁⠎ ⠠⠁ ⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝
⠠⠉⠓⠁⠗⠞⠑⠗ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠊⠝⠉⠕⠗⠏⠕⠗⠁⠞⠑⠙ ⠠⠕⠝ ⠠⠙⠑⠉⠑⠍⠃⠑⠗ ⠼⠁⠁⠂ ⠼⠁⠛⠛⠼
⠠⠃⠽ ⠠⠞⠓⠑ ⠠⠎⠞⠁⠞⠑ ⠠⠕⠋ ⠠⠙⠑⠇⠁⠺⠁⠗⠑ ⠠⠁⠎ ⠠⠁
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠞⠓⠑ ⠄⠄⠄ ⠠⠺⠁⠎ ⠠⠁⠏⠏⠕⠊⠝⠞⠑⠙ ⠠⠁⠎ ⠠⠁ ⠠⠙⠊⠗⠑⠉⠞⠕⠗ ⠕⠋ ⠠⠞⠓⠑
⠠⠉⠕⠗⠏⠕⠗⠁⠞⠊⠕⠝ ⠄⠄⠄

⠠⠊⠎⠎⠝ ⠠⠊⠎⠎⠝ ⠠⠏⠗⠊⠝⠞ ⠠⠍⠑⠙⠊⠁ ⠠⠁⠎ ⠠⠁ ⠠⠏⠗⠊⠝⠞⠑⠙ ⠠⠍⠁⠞⠞⠑⠗
⠠⠍⠁⠝⠥⠎⠉⠗⠊⠏⠞ ⠠⠋⠕⠗ ⠠⠁ ⠠⠏⠥⠃⠇⠊⠉⠁⠞⠊⠕⠝ ⠠⠊⠎⠎⠥⠑ ⠠⠃⠽ ⠠⠍⠁⠗⠅ ⠠⠎⠞⠑⠏⠓⠑⠝⠎⠕⠝

⠠⠊⠎⠎⠝ ⠠⠊⠎⠎⠝ ⠠⠍⠁⠝⠥⠎⠉⠗⠊⠏⠞ ⠠⠋⠕⠗ ⠠⠁ ⠠⠏⠥⠃⠇⠊⠉⠁⠞⠊⠕⠝ ⠠⠊⠎⠎⠥⠑ ⠠⠃⠽
⠠⠍⠁⠗⠅ ⠠⠎⠞⠑⠏⠓⠑⠝⠎⠕⠝ ⠠⠁ ⠠⠏⠗⠊⠝⠞⠑⠙ ⠠⠍⠁⠞⠞⠑⠗ ⠠⠁⠎ ⠠⠁ ⠠⠏⠥⠃⠇⠊⠉⠁⠞⠊⠕⠝
⠠⠊⠎⠎⠥⠑ ⠠⠃⠽ ⠠⠍⠁⠗⠅ ⠠⠎⠞⠑⠏⠓⠑⠝⠎⠕⠝ ⠠⠁ ⠠⠏⠗⠊⠝⠞⠑⠙ ⠠⠍⠁⠞⠞⠑⠗

⠠⠠⠠
⠠⠠⠠

⠠⠊⠝⠞⠑⠗⠝⠁⠞⠊⠕⠝⠁⠇ ⠠⠍⠁⠗⠊⠞⠊⠍⠑ ⠠⠕⠗⠛⠁⠝⠊⠵⠁⠞⠊⠕⠝ ⠠⠋⠕⠗ ⠠⠎⠞⠁⠝⠙⠁⠗⠙⠎
⠠⠁⠝⠙ ⠠⠉⠕⠝⠙⠊⠞⠊⠕⠝ ⠠⠋⠁⠉⠊⠇⠊⠞⠊⠑⠎ ⠠⠁⠗⠕⠥⠝⠙ ⠠⠞⠓⠑ ⠠⠺⠕⠗⠇⠙
⠠⠏⠁⠗⠞ ⠠⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠊⠠⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑
⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞
⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠊⠠⠊⠊⠊ ⠠⠁⠗⠞⠊⠉⠇⠑ ⠠⠁⠝⠙ ⠠⠏⠁⠗⠞ ⠠⠠⠠⠠⠠⠠⠠⠠⠠

⠠⠊⠝⠙⠑⠭ ⠕⠋ ⠠⠇⠁⠺⠎

⠠⠞⠓⠑ ⠠⠉⠕⠝⠎⠞⠊⠞⠥⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠙⠕⠑⠎ ⠝⠕⠞ ⠁⠥⠞⠓⠕⠗⠊⠵⠑ ⠁⠝⠽ ⠎⠞⠁⠞⠑ ⠕⠗ ⠞⠑⠗⠗⠊⠞⠕⠗⠽ ⠕⠗ ⠁⠝⠽ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠕⠗ ⠁⠛⠑⠝⠉⠽ ⠕⠋ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠞⠕ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠋⠕⠗ ⠁ ⠉⠕⠍⠍⠑⠗⠉⠊⠁⠇ ⠏⠥⠗⠏⠕⠎⠑ ⠊⠝ ⠞⠓⠑ ⠥⠝⠊⠞⠑⠙ ⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠁⠍⠑⠗⠊⠉⠁ ⠞⠓⠁⠞ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠓⠁⠎ ⠞⠕ ⠕⠃⠞⠁⠊⠝ ⠁ ⠇⠊⠉⠑⠝⠎⠑ ⠋⠕⠗ ⠞⠓⠁⠞ ⠏⠥⠗⠏⠕⠎⠑ ⠋⠗⠕⠍ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠠⠉⠕⠇⠕⠗⠁⠙⠕ ⠊⠝ ⠞⠓⠑ ⠉⠕⠥⠝⠞⠽ ⠕⠋ ⠠⠙⠑⠝⠧⠑⠗ ⠁⠝⠙ ⠞⠓⠑ ⠠⠙⠊⠎⠞⠗⠊⠉⠞ ⠕⠋ ⠠⠉⠕⠇⠥⠍⠃⠊⠁ ⠊⠝ ⠞⠓⠑ ⠉⠕⠥⠝⠞⠽ ⠕⠋ ⠠⠙⠑⠝⠧⠑⠗ ⠁⠝⠙ ⠞⠓⠑ ⠠⠙⠊⠎⠞⠗⠊⠉⠞ ⠕⠋ ⠠⠉⠕⠇⠥⠍⠃⠊⠁ ⠁⠗⠑ ⠓⠑⠗⠑⠃⠽ ⠙⠊⠎⠎⠕⠇⠧⠑⠙ ⠁⠝⠙ ⠁⠇⠇ ⠉⠕⠍⠏⠁⠝⠽ ⠉⠕⠝⠉⠑⠗⠝⠊⠝⠛ ⠞⠓⠑ ⠉⠕⠝⠙⠥⠉⠞ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠕⠋ ⠠⠏⠓⠽⠎⠊⠉⠊⠎⠞⠎ ⠁⠝⠙ ⠠⠁⠎⠞⠗⠕⠝⠕⠍⠑⠗⠎ ⠊⠝ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠁⠝⠙ ⠞⠕ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠋⠕⠗ ⠁ ⠉⠕⠍⠍⠑⠗⠉⠊⠁⠇ ⠏⠥⠗⠏⠕⠎⠑ ⠊⠝ ⠞⠓⠑ ⠥⠝⠊⠞⠑⠙ ⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠁⠍⠑⠗⠊⠉⠁ ⠞⠓⠁⠞ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠓⠁⠎ ⠞⠕ ⠕⠃⠞⠁⠊⠝ ⠁ ⠇⠊⠉⠑⠝⠎⠑ ⠋⠕⠗ ⠞⠓⠁⠞ ⠏⠥⠗⠏⠕⠎⠑ ⠋⠗⠕⠍ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠠⠉⠕⠇⠕⠗⠁⠙⠕ ⠊⠝ ⠞⠓⠑ ⠉⠕⠥⠝⠞⠽ ⠕⠋ ⠠⠙⠑⠝⠧⠑⠗ ⠁⠝⠙ ⠞⠓⠑ ⠠⠙⠊⠎⠞⠗⠊⠉⠞ ⠕⠋ ⠠⠉⠕⠇⠥⠍⠃⠊⠁ ⠁⠗⠑ ⠓⠑⠗⠑⠃⠽ ⠙⠊⠎⠎⠕⠇⠧⠑⠙ ⠁⠝⠙ ⠁⠇⠇ ⠉⠕⠍⠏⠁⠝⠽ ⠉⠕⠝⠉⠑⠗⠝⠊⠝⠛ ⠞⠓⠑ ⠉⠕⠝⠙⠥⠉⠞ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠕⠋ ⠠⠏⠓⠽⠎⠊⠉⠊⠎⠞⠎ ⠁⠝⠙ ⠠⠁⠎⠞⠗⠕⠝⠕⠍⠑⠗⠎ ⠊⠝ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎

⠠⠞⠓⠑ ⠠⠉⠕⠝⠎⠞⠊⠞⠥⠞⠊⠕⠝ ⠕⠋ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠙⠕⠑⠎ ⠝⠕⠞ ⠁⠥⠞⠓⠕⠗⠊⠵⠑ ⠁⠝⠽ ⠎⠞⠁⠞⠑ ⠕⠗ ⠞⠑⠗⠗⠊⠞⠕⠗⠽ ⠕⠗ ⠁⠝⠽ ⠙⠑⠏⠁⠗⠞⠍⠑⠝⠞ ⠕⠗ ⠁⠛⠑⠝⠉⠽ ⠕⠋ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠞⠕ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠠⠁⠍⠑⠗⠊⠉⠁ ⠋⠕⠗ ⠁ ⠉⠕⠍⠍⠑⠗⠉⠊⠁⠇ ⠏⠥⠗⠏⠕⠎⠑ ⠊⠝ ⠞⠓⠑ ⠥⠝⠊⠞⠑⠙ ⠎⠞⠁⠞⠑⠎ ⠕⠋ ⠁⠍⠑⠗⠊⠉⠁ ⠞⠓⠁⠞ ⠞⠓⠑ ⠉⠕⠍⠏⠁⠝⠽ ⠓⠁⠎ ⠞⠕ ⠕⠃⠞⠁⠊⠝ ⠁ ⠇⠊⠉⠑⠝⠎⠑ ⠋⠕⠗ ⠞⠓⠁⠞ ⠏⠥⠗⠏⠕⠎⠑ ⠋⠗⠕⠍ ⠞⠓⠑ ⠎⠞⠁⠞⠑ ⠕⠋ ⠠⠉⠕⠇⠕⠗⠁⠙⠕ ⠊⠝ ⠞⠓⠑ ⠉⠕⠥⠝⠞⠽ ⠕⠋ ⠠⠙⠑⠝⠧⠑⠗ ⠁⠝⠙ ⠞⠓⠑ ⠠⠙⠊⠎⠞⠗⠊⠉⠞ ⠕⠋ ⠠⠉⠕⠇⠥⠍⠃⠊⠁ ⠁⠗⠑ ⠓⠑⠗⠑⠃⠽ ⠙⠊⠎⠎⠕⠇⠧⠑⠙ ⠁⠝⠙ ⠁⠇⠇ ⠉⠕⠍⠏⠁⠝⠽ ⠉⠕⠝⠉⠑⠗⠝⠊⠝⠛ ⠞⠓⠑ ⠉⠕⠝⠙⠥⠉⠞ ⠕⠋ ⠞⠓⠑ ⠠⠁⠍⠑⠗⠊⠉⠁⠝ ⠠⠁⠎⠎⠕⠉⠊⠁⠞⠊⠕⠝ ⠕⠋ ⠠⠏⠓⠽⠎⠊⠉⠊⠎⠞⠎ ⠁⠝⠙ ⠠⠁⠎⠞⠗⠕⠝⠕⠍⠑⠗⠎ ⠊⠝ ⠞⠓⠑ ⠠⠥⠝⠊⠞⠑⠙ ⠠⠎⠞⠁⠞⠑⠎

⠠⠞⠓⠑ ⠠⠞⠗⠁⠙⠑⠍⠁⠗⠅ ⠠⠁⠛⠗⠑⠑⠍⠑⠝⠞ ⠠⠞⠓⠊⠎ ⠠⠁⠛⠗⠑⠑⠍⠑⠝⠞ ⠊⠎ ⠍⠁⠙⠑ ⠁⠝⠙ ⠑⠝⠞⠑⠗⠑⠙ ⠊⠝⠞⠕ ⠃⠽ ⠁⠝⠙ ⠃⠑⠞⠺⠑⠑⠝ ⠞⠓⠑ ⠠⠁⠥⠞⠓⠕⠗ ⠁⠝⠙ ⠠⠁⠛⠗⠑⠑⠍⠑⠝⠞ ⠁⠝⠙ ⠞⠓⠑ ⠠⠁⠥⠞⠓⠕⠗ ⠁⠝⠙ ⠠⠁⠛⠗⠑⠑⠍⠑⠝⠞ ⠁⠗⠑ ⠝⠕⠞ ⠑⠍⠏⠇⠕⠽⠑⠑⠎ ⠕⠋ ⠞⠓⠑ ⠠⠁⠥⠞⠓⠕⠗ ⠁⠝⠙ ⠠⠁⠛⠗⠑⠑⠍⠑⠝⠞ ⠁⠗⠑ ⠝⠕⠞ ⠁⠋⠋⠊⠇⠊⠁⠞⠑⠙ ⠺⠊⠞⠓ ⠁⠝⠽

⠠⠠⠠
⠠⠠⠠

⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠙⠑ ⠊⠝⠊⠉⠊⠕ ⠙⠑ ⠍⠁⠗⠵⠕ ⠎⠑ ⠍⠁⠝⠞⠥⠧⠕
⠕ ⠍⠕⠙⠑⠗⠁⠙⠁ ⠞⠁⠍⠃⠊⠑́⠝ ⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠉⠕⠝⠎⠞⠁⠝⠞⠑⠍⠑⠝⠞⠑
⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑. ⠠⠑⠇ ⠉⠁⠍⠃⠊⠕ ⠙⠑ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠙⠑⠍⠁⠝⠙⠁
⠇⠁ ⠉⠁⠇⠙⠁ ⠁ ⠇⠁ ⠎⠁⠇⠊⠙⠁ ⠙⠑ ⠇⠁ ⠉⠁⠇⠑⠋⠁⠉⠉⠊⠕́⠝ ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁
⠍⠁⠭⠊⠍⠁ ⠙⠑ ⠇⠁ ⠉⠁⠇⠙⠁ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠍⠁⠭⠊⠍⠁
⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎ ⠞⠗⠑⠎
⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠍⠊⠝⠊⠍⠁
⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠍⠁⠭⠊⠍⠁
⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑.

⠠⠑⠇ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁
⠙⠑ ⠇⠁⠎ ⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁
⠍⠁⠭⠊⠍⠁ ⠙⠑ ⠇⠁ ⠉⠁⠇⠙⠁ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠍⠊⠝⠊⠍⠁
⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠍⠁⠭⠊⠍⠁
⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑.

⠠⠑⠇ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁
⠙⠑ ⠇⠁⠎ ⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁
⠍⠁⠭⠊⠍⠁ ⠙⠑ ⠇⠁ ⠉⠁⠇⠙⠁ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠍⠊⠝⠊⠍⠁
⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠍⠁⠭⠊⠍⠁
⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑. ⠠⠇⠁ ⠞⠑⠍⠏⠑⠗⠁⠞⠥⠗⠁ ⠍⠊⠝⠊⠍⠁
⠙⠑ ⠇⠁ ⠝⠕⠉⠓⠑ ⠎⠑ ⠁⠇⠉⠁⠝⠵⠁ ⠑⠝ ⠇⠁ ⠎⠑⠍⠁⠝⠁ ⠙⠑ ⠇⠁⠎
⠞⠗⠑⠎ ⠽ ⠍⠑⠙⠊⠁ ⠙⠑ ⠇⠁ ⠞⠁⠙⠑.

⠈⠞⠗⠁ ⠍⠁⠝⠽ ⠙ ⠍⠊⠇⠇⠊⠕⠝⠎ ⠍⠊ ⠈⠗ ⠍⠈⠇⠞⠊⠍⠈⠗⠙⠎ ⠍⠈ ⠈⠗⠁
⠞ ⠙ ⠙⠍ ⠈⠞⠗⠁⠝⠎⠏⠕⠗⠞⠁⠞⠊⠕⠝ ⠍⠎ ⠈⠗⠁⠍⠞ ⠙ ⠍⠈ ⠇⠕⠉⠁⠇⠎⠞⠗⠁⠇
⠞⠎ ⠍⠊ ⠞⠈⠞⠁⠇ ⠃⠇⠕⠉⠅⠎ ⠍⠕⠗⠞⠁⠇⠊⠞⠽ ⠙ ⠃⠇⠁⠉⠅
⠈⠇⠇⠕⠉⠁⠞⠊⠕⠝ ⠍⠈ ⠈⠗⠁⠍⠞ ⠙ ⠍⠊⠞ ⠁ ⠃⠁⠙ ⠉⠕⠙⠑ ⠙ ⠈⠞
⠞⠈⠞⠁ ⠍⠈ ⠈⠞⠗⠁⠝⠎⠁⠉⠞⠊⠕⠝⠎ ⠙⠍ ⠊⠝⠙⠑⠏⠑⠝⠙⠑⠝⠞ ⠙ ⠍⠎
⠃⠇⠕⠉⠅⠎ ⠍⠎ ⠁⠝⠙ ⠙ ⠍⠊⠇⠇⠊⠕⠝⠎ ⠍⠈ ⠁⠗⠑ ⠝⠕⠞ ⠁
⠃⠇⠕⠉⠅⠑⠙ ⠍⠎ ⠁⠝⠙ ⠍⠕⠗⠞⠁⠇⠊⠞⠽ ⠙ ⠁⠉⠉⠕⠥⠝⠞⠁⠃⠇⠑
⠈⠞⠗⠁⠝⠎⠁⠉⠞⠊⠕⠝⠎ ⠍⠎ ⠁⠗⠑ ⠙⠕⠝⠑ ⠁⠞ ⠞⠓⠑ ⠁

⠠⠁⠇⠇ ⠗⠊⠛⠓⠞⠎ ⠗⠑⠎⠻ ⠠⠞⠓⠑ ⠠⠁⠥⠞⠓⠕⠗
⠠⠁⠝ ⠠⠓⠊⠎⠞⠕⠗⠽ ⠠⠞⠓⠑ ⠠⠃⠕⠕⠅ ⠕⠋ ⠠⠚⠕
⠃⠎. ⠠⠠ ⠠⠠⠁⠠⠠⠙ ⠙⠕⠍⠊⠝⠊⠕⠝ ⠠⠠
⠠⠠⠠ ⠠⠠⠠⠠⠙ ⠠⠓⠊⠎⠞⠕⠗⠽ ⠠⠠⠠ ⠠⠠⠠
⠠⠠⠠⠠. ⠠⠠⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠
⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠⠠⠠