

Campeonato Nakahata de Programação (Chave Ouro)

Dificuldade Fácil (1 ponto cada)

1 - Contando palíndromos - Dada uma sequência de strings, conte quantas dessas strings são palíndromos. Um palíndromo é uma palavra que lê o mesmo para frente e para trás.

Entrada:

- Uma sequência de N strings.

Saída:

- Um número inteiro representando a quantidade de palíndromos na sequência.

Exemplos:

Entrada: radar, level, nakahata, madam

Saída: 3

Entrada: odeio, programar, racecar

Saída: 1

Entrada: arara, macaco, tomate, rotor

Saída: 2

2 - Contando 1's - Dado um número inteiro N, você precisa encontrar quantos números de 1's estão presentes na representação binária de N.

Entrada:

- Um número inteiro N ($1 \leq N \leq 10000$) indicando o número que você encontrará os 1's.

Saída:

- Um número inteiro representando a quantidade de 1's na representação binária de N.

Exemplos:

Entrada: 13 (Representação binária: 1101)

Saída: 3

Entrada: 10 (Representação binária: 1010)

Saída: 2

Entrada: 999 (Representação binária: 1111100111)

Saída: 8

3 – Soma de subconjuntos - Dado um conjunto de números inteiros e um número inteiro S, determine se existe um subconjunto dos números que soma exatamente S.

Entrada:

- Uma sequência de N números inteiros, que representam os elementos do conjunto.
- Um número inteiro S representando a soma desejada do subconjunto.

Saída:

- Um valor booleano (1 para verdadeiro, 0 para falso) indicando se existe um subconjunto cujo soma é exatamente S.

Exemplos:

Entrada:

3 3 4 4 12

9

Saída: 0

Descrição do resultado: Não existe subconjunto que soma 9, saída falsa (0).

Entrada:

2 2 3 7 8

10

Saída: 1

Descrição do resultado: O subconjunto que soma 10 é [3, 7], saída verdadeira (1).

Entrada:

1 2 5

7

Saída: 1

Descrição do resultado: O subconjunto que soma 7 é [2, 5], saída verdadeira (1).

4 – Soma dos números primos - Dado um conjunto de números inteiros, calcule a soma de todos os números primos presentes no conjunto.

Entrada:

- Uma sequência de N números inteiros.

Saída:

- Um número inteiro representando a soma de todos os números primos na sequência.

Exemplos:

Entrada: 3 4 7 10 13

Saída: 23

Entrada: 6 8 10 12

Saída: 0

Entrada: 3 4 7 10 13

Saída: 23

Entrada: 2 3 5 7 11 13 17

Saída: 58

5 – Verificando Anagramas - Dado duas palavras, determine se essas palavras anagramas entre si. Duas palavras são consideradas anagramas se elas contiverem exatamente as mesmas letras, mas em ordens diferentes.

Entrada:

- Duas palavras em letras minúsculas separada por um espaço.

Saída:

- Um valor booleano (1 para verdadeiro, 0 para falso).

Entrada: amor roma

Saída: 1.

Entrada: mato tomate

Saída: 0.

Entrada: treco corte

Saída: 1.

Dificuldade Média (3 pontos cada)

6 – Somando pares de números - Dado um conjunto de números inteiros, determine quantos pares distintos de números têm uma soma que é um número par.

Entrada:

- Uma sequência de N números inteiros, onde $1 \leq N \leq 1000$.

Saída:

- Um número inteiro representando a quantidade de pares distintos cuja soma é um número par.

Entrada: 1 2 3 4 5

Saída: 4

Descrição do resultado: (1, 3), (1, 5), (2, 4), (3, 5)

Entrada: 1 1 1 2 2

Saída: 4

Descrição do resultado: (1, 1), (1, 1), (1, 1), (2, 2)

Entrada: 2 2 2 2

Saída: 6

Descrição do resultado: (2, 2), (2, 2), (2, 2), (2, 2), (2, 2), (2, 2)

Informações Adicionais:

- Considere que a ordem dos pares não importa. Ou seja, (a, b) é o mesmo que (b, a).
- Um par é considerado distinto se os **índices dos números que o compõem são diferentes**, mesmo que os números sejam iguais.

7 – Sequência mais longa - Dada uma sequência de números inteiros, encontre a subsequência crescente mais longa. A subsequência **não precisa ser contínua**, mas deve ser a **mais longa**

possível. Uma subsequência é uma sequência derivada removendo alguns (ou nenhum) elementos da sequência original sem mudar a ordem dos elementos restantes.

Entrada:

- Uma sequência de N números inteiros.

Saída:

- O comprimento da subsequência crescente mais longa.

Exemplos:

Entrada: 10 22 9 33 21 50

Saída: 4

Descrição do resultado: A subsequência crescente mais longa é [10, 22, 33, 50] e tem comprimento 4.

Entrada: 3 10 2 1 20

Saída: 3

Descrição do resultado: A subsequência crescente mais longa é [3, 10, 20] e tem comprimento 3.

Entrada: 50 3 10 7 40 80

Saída: 4

Descrição do resultado: A subsequência crescente mais longa é [3, 10, 40, 80] e tem comprimento 4 **ou** [3, 7, 40, 80] e tem comprimento 4.

8 - Knapsack Problem - Dado um conjunto de itens, cada um com um valor e um peso, e uma mochila com uma capacidade máxima de peso, sua tarefa é determinar o conjunto de itens que maximiza o valor total, sem exceder a capacidade máxima da mochila.

Entrada:

- **Capacidade da Mochila:** Um número inteiro C representando a capacidade máxima da mochila em unidades de peso. ($1 \leq C \leq 1000$).
- **Itens:** Uma lista de n itens, onde cada item é representado por dois números inteiros w_i e v_i :
 - w_i representa o peso do item i. ($1 \leq w_i \leq 1000$)
 - v_i representa o valor do item i. ($1 \leq v_i \leq 1000$)

Saída:

Um número inteiro representando o valor máximo que pode ser obtido com a mochila, sem exceder sua capacidade.

Exemplos:

Entrada:

5
2 3
3 4
4 5
5 6

Saída: 7

Descrição da entrada:

1º linha: Capacidade da Mochila: 5

2º linha: Item 1: Peso = 2, Valor = 3

3º linha: Item 2: Peso = 3, Valor = 4

4º linha: Item 3: Peso = 4, Valor = 5

5º linha: Item 4: Peso = 5, Valor = 6

Entrada:

10

5 10

4 40

6 30

Saída:

70

Descrição da entrada:

1º linha: Capacidade da Mochila: 10

2º linha: Item 1: Peso = 5, Valor = 10

3º linha: Item 2: Peso = 4, Valor = 40

4º linha: Item 3: Peso = 6, Valor = 30

Dificuldade Difícil (5 pontos)

9 - Evitando confusão – Dado um conjunto de coordenadas representando a posição de alunos em um plano cartesiano 2D, sua tarefa é calcular a quantidade máxima de alunos que podem ser colocados juntos em uma sala de aula, de modo que a distância entre qualquer par de alunos seja maior que 5 metros.

Entrada:

- Um conjunto de pares de coordenadas inteiras (x,y) representando a posição dos alunos no plano cartesiano 2D.
- A entrada será composta por um conjunto de coordenadas por linha.

Saída:

- Para cada conjunto de coordenadas, imprima um número inteiro representando a quantidade máxima de alunos que podem ser colocados na sala de aula, de forma que a distância entre qualquer par de alunos seja maior que 5 metros.

Exemplos:

Entrada: {{1, 2}, {2, 4}}

Saída: 1

Entrada: {{1, 2}, {2, 4}, {3, 5}, {4, 7}, {8, 9}}

Saída: 2

Entrada: {{0, 0}, {0, 6}, {6, 0}, {6, 6}}
Saída: 4

Informações Adicionais:

- **Cálculo de Distância Euclidiana:** A distância d entre dois pontos (x_1, y_1) e (x_2, y_2) no plano cartesiano 2D pode ser calculada usando a fórmula da distância euclidiana:

$$d(Euc) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Preenchimento após finalização da prova.

Nome	Tempo Inicio	Tempo Terminou

	Sinalize os exercícios feitos	Parecer comissão organizadora
1		
2		
3		
4		
5		
6		
7		
8		
9		

Eu, _____, declaro que:

1. Concordo em participar do Campeonato Nakahata de Programação, conforme as regras e regulamentos estabelecidos para o evento.
2. Comprometo-me a respeitar todas as decisões tomadas pela comissão organizadora durante o campeonato, inclusive em relação à avaliação dos exercícios e à atribuição de pontos.
3. Declaro que não irei recorrer de qualquer decisão da comissão organizadora e aceito que todas as decisões são finais.
4. Estou ciente de que a comissão organizadora reserva-se o direito de desclassificar qualquer participante que não cumprir as regras estabelecidas ou que utilize meios inadequados para obter vantagem.

Por meio deste documento, afirmo que li, compreendi e concordo com todos os termos e condições descritos acima.

Apucarana, 23 de Agosto de 2024