



- Os exercícios da Seção 1 valem nota apenas se entregues dentro do prazo.
- Os exercícios da Seção 2, se entregues, podem valer bônus ao fim da disciplina.
- Submeta ao tidia um único arquivo .zip, que deve conter um único diretório com as soluções dos exercícios que você resolveu, devidamente nomeados (utilize o padrão indicado em negrito no enunciado). Na dúvida, veja os slides da aula 1.
- Critério de correção de cada exercício: 0 se incorreto, 0.75 se correto mas não segue os padrões de submissão, ou 1 se correto e segue os padrões.
- **Não utilize** conceitos que ainda não foram aprendidos em sala de aula: essa lista visa o treinamento dos comandos `while`, `do-while` e `for`.

## 1 Exercícios

**QtddDivisores** Faça um programa que imprima a quantidade de divisores de um número.

**Entrada:** O programa deve receber um único número inteiro  $n$ , com  $n \geq 1$ .

**Saída:** A resposta consistirá em uma única linha, que contém a quantidade total de divisores de  $n$ .

**PrimeirosPrimos** Faça um programa que imprima os  $n$  primeiros números primos.

**Entrada:** O programa deverá receber um único inteiro  $n$ , com  $n \geq 0$ .

**Saída:** A resposta consistirá em  $n$  linhas, cada uma contendo um número primo.

**Perfeitos** Um número inteiro positivo é dito perfeito se a soma dos seus divisores positivos (excluindo ele mesmo) é igual ao próprio número. Por exemplo, 6 é perfeito pois  $1 + 2 + 3 = 6$ . Faça um programa que determine se um dado número é perfeito.

**Entrada:** O programa deve receber um único número inteiro  $n$ , com  $n \geq 1$ .

**Saída:** A resposta consistirá de uma única linha, contendo a palavra “SIM”, se  $n$  for perfeito, e “NAO”, caso contrário.

**MediaPonderada** Faça um programa que calcule a média ponderada de  $n$  números de acordo com a fórmula

$$\frac{\sum_{i=1}^n p_i x_i}{\sum_{i=1}^n p_i}.$$

Não use vetores.

**Entrada:** O programa deve receber inicialmente um inteiro positivo  $n$ . Em seguida, deve receber  $n$  pares de números reais  $p_i$  e  $x_i$ .

**Saída:** A resposta consistirá de uma única linha, contendo apenas um valor real com 2 casas decimais que é a média ponderada dos valores lidos, de acordo com a fórmula acima.

**PrimosProximos** Escreva um programa que mostre os dois números primos mais próximos de um dado número  $n$ . Por exemplo, se  $n = 24$ , os dois números primos mais próximos dele são 23 e 29. Por outro lado, se  $n = 47$ , então os dois números primos mais próximo dele são 47 e 53.

**Entrada:** O programa deve receber um número inteiro positivo  $n \geq 3$ .

**Saída:** A resposta consistirá de uma única linha contendo dois números inteiros separados por espaço onde o primeiro dos números é o maior número primo que é menor ou igual a  $n$  e o segundo dos números é o menor número primo que é maior do que  $n$ .

**Desenho** Faça um programa que imprima uma figura similar à seguinte, mas com  $n$  linhas.

```
...*...
..***..
.*****.
*****
```

**Entrada:** O programa deve receber um único número inteiro  $n$ , com  $n \geq 0$ .

**Saída:** A resposta consistirá em  $n$  linhas, conforme o padrão desenhado acima.

**Equacao** Faça um programa que calcule todas as soluções inteiras de equações do tipo

$$x_1 + x_2 + x_3 + x_4 = C$$

onde  $x_1 \geq 0$ ,  $x_2 \geq 0$ ,  $x_3 \geq 0$ ,  $x_4 \geq 0$  e  $C \geq 0$ .

**Entrada:** O programa deve receber um único número inteiro  $C \geq 0$ .

**Saída:** A resposta consistirá de zero ou mais linhas, onde cada linha contém uma solução da equação: 4 inteiros separados por espaço representando os valores de  $x_1$ ,  $x_2$ ,  $x_3$  e  $x_4$ .

**Exemplo:** Se  $C = 4$ , então a saída deve ser

```
0 0 0 4
0 0 1 3
0 0 2 2
0 0 3 1
0 0 4 0
0 1 0 3
0 1 1 2
0 1 2 1
0 1 3 0
0 2 0 2
0 2 1 1
0 2 2 0
```

```

0 3 0 1
0 3 1 0
0 4 0 0
1 0 0 3
1 0 1 2
1 0 2 1
1 0 3 0
1 1 0 2
1 1 1 1
1 1 2 0
1 2 0 1
1 2 1 0
1 3 0 0
2 0 0 2
2 0 1 1
2 0 2 0
2 1 0 1
2 1 1 0
2 2 0 0
3 0 0 1
3 0 1 0
3 1 0 0
4 0 0 0

```

**Palindromo** Dado um número inteiro não negativo  $n$ , seja  $\bar{n}$  o número que se obtém invertendo-se a ordem dos dígitos de  $n$ . Por exemplo, se  $n = 332$ , então  $\bar{n} = 233$ . Um número é um palíndromo se  $\bar{n} = n$ . Por exemplo, 34543, 1 e 99 são palíndromos. Escreva um programa que verifique se um número é palíndromo ou não.

**Entrada:** O programa deve receber um único número inteiro  $n$ , com  $n \geq 0$ .

**Saída:** A resposta consistirá de uma única linha, contendo a palavra “SIM”, se  $n$  for palíndromo, e “NAO”, caso contrário.

**Binario** Converter um inteiro positivo para sua representação binária. Por exemplo, se o inteiro é 34, sua representação binária correspondente é “100010”, pois  $1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 34$ . Não é permitido usar funções da biblioteca do Java.

**Entrada:** O programa deve receber um único número inteiro  $n$ , com  $n \geq 0$ .

**Saída:** A resposta consistirá de uma única linha, contendo a representação binária de  $n$ .

## 2 Bônus

**Crescente** Faça um programa que teste se uma sequência de números é crescente ou não. Não use vetores para armazenar os números.

**Entrada:** O programa deve receber  $n$  números inteiros.

**Saída:** A resposta consistirá de uma única linha, contendo a palavra “SIM”, se a sequência de números for crescente, e “NAO”, caso contrário.