

Subprogramas

1. **Verifique** se um dado número inteiro positivo é **par** ou **ímpar**. Para tal, implemente o subprograma **parOuImpar**, cujo protótipo se apresenta a seguir, e que escreve no monitor “O número é par” ou “O número é ímpar” consoante o valor inteiro passado como parâmetro seja par ou ímpar, respectivamente:

void parOuImpar (int);

2. **Verifique** se um dado número inteiro positivo é **par**. Para tal, implemente a função **par**, cujo protótipo se apresenta a seguir, e que devolve 1 se o inteiro passado como parâmetro for par ou 0 caso contrário:

int par (int); // protótipo

3. **Verifique** se um dado número inteiro positivo é **ímpar**. Para tal, implemente a função **impar**, cujo protótipo apresenta-se a seguir, e que devolve 1 se o inteiro passado como parâmetro for ímpar ou 0 caso contrário:

int impar (int); // protótipo

4. Determine o **maior** de dois números inteiros. Para tal, deverá implementar a função **maior**, cujo protótipo se apresenta a seguir, e que devolve o maior dos dois valores inteiros passados como parâmetro:

int maior (int, int); // protótipo

5. Determine o **maior** de três números inteiros. Para tal, deverá implementar a função **maiorDeTres** que devolve o maior de três valores inteiros passados como parâmetros:

int maiorDeTres (int, int, int); // protótipo

6. Determine a **média** de dois números inteiros. Para tal, deverá implementar a função **media**, cujo protótipo se apresenta a seguir, e que devolve a media dos dois valores inteiros passados como parâmetros:

float media (int, int); // protótipo

7. Determinar o **maior** número de uma sequência de N ($N \geq 2$) números reais dados pelo utilizador (se $N < 2$, então N deverá ser pedido novamente até obter um valor válido). Deverá usar a função **maiorN**, na qual serão introduzidos os N números e determinado o maior deles:

float maiorN (int); // protótipo

8. Calcular a **soma** dos dígitos de um número inteiro positivo. Para tal, implemente a função **somaDigitos**, cujo protótipo se apresenta a seguir, e que devolve a soma dos dígitos de um dado número inteiro positivo passado como parâmetro:

int somaDigitos (int); // protótipo

9. Determinar o **maior** e o **menor** número de uma sequência de N ($N \geq 2$) números reais inseridos pelo utilizador. Inserir um valor inteiro N: se $N < 2$, então N deverá ser pedido novamente até ser inserido um valor válido. Após a introdução dos N números reais, a aplicação deverá mostrar o maior e o menor valores entre os valores inseridos. Deverá implementar e usar as funções **maximo** e **minimo** seguintes:

float maximo (float, float); // protótipo

float minimo (float, float); // protótipo

10. Determinar a **soma** e a **média** de uma sequência de N ($N \geq 2$) números reais dados pelo utilizador (se $N < 2$, então N deverá ser pedido novamente até obter um valor válido). Deverá usar a função **soma**, na qual serão introduzidos os N números e calculada a respectiva soma:

float soma (int); // protótipo

11. Mostre uma **pirâmide** de números como a que se segue. Para tal, deverá pedir ao utilizador o valor de N maior ou igual 2. Para $N = 5$, a pirâmide é a seguinte:

```

                1
              1 2 1
            1 2 3 2 1
          1 2 3 4 3 2 1
        1 2 3 4 5 4 3 2 1
```

Deverá implementar os subprogramas, com os protótipos que se seguem, para:

→ escreve o carácter associado ao primeiro parâmetro, tantas vezes quantas as do valor associado ao segundo parâmetro

void escreveCaracter (char, int); // protótipo

→ mostra os números inteiros entre os valores associados ao primeiro e ao segundo parâmetros em ordem crescente:

void mostraNumerosAsc (int, int); // protótipo

→ mostra os números inteiros entre os valores associados ao primeiro e ao segundo parâmetros em ordem decrescente:

void mostraNumerosDesc (int, int); // protótipo