

## TESTE DE AVALIAÇÃO FINAL

- 1 Uma sequência finita e não ambígua de instruções de modo a obter a resolução de um problema sob a forma de resultado (saída de dados) tendo por base uma prévia entrada de dados. A descrição refere-se a:

- a) Objetos;
- b) Algoritmo;
- c) Compilador;
- d) Modularização.

R: b

- 2 Os algoritmos são compostos por três tipos de instrução, de sequência, de decisão e de repetição. Que tipos de instruções encontra no seguinte excerto de código.

```
Console.Write("Nome: ");  
nome = Console.ReadLine();  
Console.Write("Idade: ");  
idade = int.Parse(Console.ReadLine());  
  
Console.Clear();  
  
if (idade > 18)  
    Console.WriteLine($"{nome} é maior de 18 anos");
```

- a) sequência e decisão;
- b) sequência e repetição;
- c) decisão e repetição;
- d) todas.

R: a

- 3 Pretende-se uma aplicação que escreva os números de 1 até 100. Indique o bloco de repetição mais indicado para implementar a solução para o problema.

- a) `while()`;
- b) `do{}while()`;
- c) `for()`.

R: c

- 4 Analise a seguinte afirmação:

Um procedimento retorna sempre um tipo de dados de retorno associado, enquanto que uma função não devolve qualquer valor.

- a) a afirmação é verdadeira;
- b) a afirmação é falsa, quer um procedimento quer uma função, não tem associado qualquer tipo de dados de retorno;
- c) a afirmação é falsa, quer um procedimento quer uma função retorna sempre um tipo de dados de retorno associado;
- d) a afirmação é falsa, uma função tem sempre um tipo de dados de retorno associado, enquanto que um procedimento não retorna qualquer valor.

R: d

- 5 A cidade de Lisboa pretende impor limites à circulação automóvel no centro da cidade. Pretende-se implementar um algoritmo que leia o último número da matrícula de um automóvel e que mostra a mensagem de acordo com o dia da semana que este automóvel não pode circular.

```
0-2  "Não Circular segunda-feira"  
3    "Não Circular terça-feira"  
4    "Não Circular quarta-feira"  
5    "Não Circular quinta-feira"  
6    "Não Circular sexta-feira"
```

Qual o bloco de instrução de decisão mais adequado a usar na implementação da solução deste problema?

- a) `if()..if()..else` ou `if()..else if()else()`;
- b) `if()..else if()else()` ou `switch() case`;
- c) `switch() case` ou `if()..if()..else`.

R: b

- 6 Analise o seguinte excerto de código:

```
bool a = false, b = true, c = false;  
  
a = a || b && c && !a;  
b = (b && !c) || (a && b);  
c = !a && !c && b;
```

Qual o resultado final de cada uma das variáveis?

- a) `a = false, b = true, c = false;`
- b) `a = false, b = false, c = true;`
- c) `a = false; b = true; c = true;`
- d) `a = true, b = true, c = true;`

R: d

7 Algumas das vantagens do uso de sub-rotinas são:

- a) reutilização de código, maior rapidez de processamento e facilidade de manutenção e evolução do código;
- b) otimiza o trabalho em equipa, promove a reutilização de código e gestão de memória mais eficiente;
- c) otimiza o trabalho em equipa, maior rapidez de processamento, reutilização de código;
- d) maior rapidez de processamento, promove a reutilização de código e gestão de memória mais eficiente.

R: b

8 Analise o seguinte excerto de código

```
int a = 16, b = 4;  
  
if (a + b >= 20)  
    a = a / b;  
else  
    a = a % b;
```

Indique o resultado final de cada uma das variáveis.

- a) a = 4, b = 4;
- b) a = 2, b = 4;
- c) a = 6, b = 4;
- d) a = 4, b = 6;

R: a

9 Analise o seguinte excerto de código:

```
int x = 3, y = 1;

do
{
    if (x % 2 != 0)
        x++;
    else
    {
        y++;
        x = x + y;
    }
} while (x < 6);
```

Indique o resultado final de cada uma das variáveis.

- a) x = 1, y = 5;
- b) x = 4, y = 8;
- c) x = 3, y = 6;
- d) x = 6, y = 2;

R: d

10 Analise o seguinte excerto de código:

```
int i, j = 1;

for (i = 0; i < 4; i++)
{
    if (i % 2 == 0)
    {
        j++;
    }
}
```

Indique o resultado final de cada uma das variáveis.

- a) x = 5, y = 4;
- b) x = 4, y = 3;
- c) x = 3, y = 3;
- d) x = 4, y = 2;

R:b