

Engenharia InformáticaFrequência - Época Normal **Algoritmos e Programação**

Ano letivo 2024/2025 – 1º Ano/1º Semestre

10.01.2025

Duração: 120 Minutos Sem Consulta

Nome: _____

N.º

--	--	--	--	--

Notas: Neste teste deve considerar que todas as secções de código pedidas têm de ser escritas na linguagem de programação C.

Todo o material fornecido pelo docente deve ser entregue no final da prova.

Não é permitida a utilização de qualquer dispositivo eletrónico.

1. (2.0V)

<p>a) O que aparecerá no ecrã depois do código abaixo? Justifique.</p> <pre>int i, vetor[4] = {1, 2, 3, 4}; for (i = 0; i < 4; i++) { vetor[i] = vetor[i] * 2; } for (i = 0; i < 4; i++) { printf("%d ", vetor[i]);}</pre> <p>a) 1 2 3 4 b) 2 4 6 8 c) 0 2 4 6 d) 1 4 9 16.</p>	<p>b) Qual o valor de j no final do ciclo? Justifique.</p> <pre>int i = 0, j = 0; while (i++ < 5) { j += i; }</pre> <p>a) 10 b) 15 c) 20 d) 21</p>
---	---

2. (2.0V) Desenvolva uma função que imprima a seguinte matriz triangular superior, para um dado número de linhas, pedido dentro da função e que não ultrapasse o valor 9.

1	2	3	4	5
0	1	2	3	4
0	0	1	2	3
0	0	0	1	2
0	0	0	0	1

exemplo para 5 linhas

3. (2.0V) Desenvolva uma função que conte e devolva o número de palavras de uma determinada string (pode considerar que as palavras estão separadas por um espaço).

```
// @param str: string de entrada
// @param nP: número de palavras
```

```
void nPalavras(char *str, int *nP)
```

4. (2.0V) Ainda em relação à função anterior (nPalavras), escreva as instruções, por exemplo no main, que lhe permitam pedir uma string ao utilizador, invocar a função nPalavras e escrever, no ecrã, a resposta da função (isto é, escrever, no ecrã, o número de palavras da string introduzida pelo utilizador).

5. Considere o seguinte excerto:

```
#define MAX_AZ1 100
#define MAX_AZ2 200
```

```
typedef struct azulejo {
    float comp;
    float larg;
    int nCores;
}AZULEJO;
```

a) (1.5V) Elabore a instrução que lhe permita definir uma parede com MAX_AZ1 azulejos de largura e MAX_AZ2 de altura.

b) (2.0V) Elabore uma função que, dada uma parede, lhe permita calcular e devolver o valor da respetiva área.

```
float areaParede(AZULEJO parede[MAX_AZ1][MAX_AZ2])
```

c) (2.0V) Elabore uma função que calcule e devolva o número médio de cores utilizado nos azulejos de uma parede.

```
float mediaCores(AZULEJO parede[MAX_AZ1][MAX_AZ2])
```

d) (2.5V) Desenvolva uma função que escreva, num ficheiro de texto, a área dos azulejos com mais do que 2 cores.

```
void infObra(AZULEJO parede[MAX_AZ1][MAX_AZ2])
```

Obs. A sintaxe de fprintf é: `int fprintf(FILE *stream, const char *format, ...);`

6. (2.0V) Complete a função abaixo, usando 4 instruções, que copie o conteúdo da string x para a string y, usando ponteiros.

```
void strCopiaComPonteiros(char *x, char *y){
    while (CONDIÇÃO)
    {
        //instr1
        // instr2
        // instr3
    }
    instr4
}
```

7. (2.0V) Considere o vetor $V=\{5,1,4,2,7,8,3,6\}$. Qual será o conteúdo do vetor V depois de executado o algoritmo abaixo?

```
INÍCIO
  Para i de 7 até 4 incremento -1 Faz
    aux <- V[i];
    V[i] <- V[7-i+1];
    V[7-i+1] <- aux;
  Fim Para
  V[2] <- V[0];
  V[V[2]] <- V[V[1]];
FIM
```