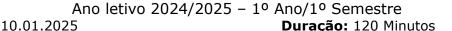
Escola Superior de Tecnologia e Gestão de Viseu

Engenharia Informática







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)

```
a) O que aparecerá no ecrã depois do código abaixo?
                                                            b) Qual o valor de j no final do ciclo? Justifique.
Justifique.
                                                            int i = 0, j = 0;
                                                            while (i++ < 5) \{ j += i; \}
int i, vetor[4] = \{1, 2, 3, 4\};
for (i = 0; i < 4; i++) {
   vetor[i] = vetor[i] * 2; }
                                                            a) 10
for (i = 0; i < 4; i++) {
                                                            b) 15
   printf("%d ", vetor[i]);}
                                                            c) 20
a) 1234
                                                            d) 21
b) 2468
c) 0 2 4 6
d) 1 4 9 16.
```

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.

```
\begin{bmatrix} 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 \end{bmatrix} 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

#define MAX_AZ2 200

float comp;
float larg;
int nCores;
}AZULEJO;
```

informática

Sem Consulta

- **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]];</pre>
FIM
```