

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)

a) Considere a função “Func” com o código abaixo. Assumindo que a função é chamada Func(“Descobre isso”, ‘s’);

Qual o valor de retorno? Justifique.

```
int Func(char *st, char car) {  
    if (!st || (*st == '\0')) return 0;  
    if (*st == car) return 1 + Func(++st, car);  
    else return Func(++st, car);  
};
```

- A. 13;
- B. 12;
- C. 3;
- D. todas as respostas anteriores estão incorretas ou existem erros.

b) Considere o seguinte código (a estrutura Pessoa está definida com vários campos):

```
int i;  
struct Pessoa *VP;  
VP = (struct Pessoa *)malloc(10*sizeof(struct Pessoa));  
//o programa continua....
```

Pretende-se destruir toda a informação alocada, qual a instrução correta?

- A. for (i=0; i<10;i++) free(VP[i]);
- B. free(VP);
- C. for (int i=0; i<10;i++){free(VP[i]); free VP;}
- D. todas as respostas anteriores estão incorretas ou existem erros.

2. Considere o somatório $\sum_{i=1}^{n+1} i * (i + 3)$.

a) (1.5V) Elabore uma função que calcule o somatório de forma iterativa.

b) (1.5V) Elabore uma função que calcule o somatório de forma recursiva.

3. (2.0V) Desenvolva uma função que receba 3 números inteiros e disponibilize os seus: máximo, média aritmética e simétrico do menor.

```
void operacoes(int n1, int n2, int n3, int* max, float* media, int* sim)
```

4. Considere o seguinte excerto:

```
#define MAX_UCS 10  
typedef struct uc {  
    char nomeUC[15];  
    float notaUC;  
} UC;
```

```
typedef struct aluno {  
    char nome[50];  
    UC cadeiras[MAX_UCS];  
    float mediaUCs;  
} Aluno;
```

a) (1.5V) Elabore as instruções que lhe permitam criar o vetor turma com um número de alunos definido pelo utilizador.

b) (2.0V) Elabore uma função que calcule e atualize, para cada aluno, o campo mediaUCs.

```
// @param turma: Vetor de alunos  
// @param n: Número de alunos no vetor anterior  
void notaMediaAlunos(Aluno *turma, int n)
```

c) (2.0V) Implemente a função `posicaoDaMelhor` que determina e devolva/disponibilize a maior média e respetiva posição/índice obtida por um aluno da turma (considere que a alínea b) foi executada).

```
// @param turma: Vetor de alunos
// @param n: Número de alunos no vetor anterior
// @param maior: Maior média
```

```
int posicaoDaMelhor(Aluno* turma, int n, float* maior)
```

d) (2.0V) Desenvolva uma função que escreva, num ficheiro de texto, o(s) nome(s) e nota(s) do(s) aluno(s) de uma determinada UC.

```
// @param turma: Vetor de alunos
// @param n: Número de alunos no vetor anterior
// @param nome: Nome da UC
// @param f: ficheiro a gravar
```

```
void infoUC(Aluno* turma, int n, char* nome, FILE *f)
```

5. (1.5V) Determine o resultado das seguintes instruções:

```
int i = -1;
printf ("%d\n", ++i);
printf ("%d", i++);
printf ("%d\n", ++i);
```

6. a) (1.5V) Diga o que aparece no ecrã, após a execução das seguintes instruções:

```
int a = 8, b = 27;
int *ptr1 = NULL, *ptr2=NULL;
ptr1 = &a;
printf("Valores de a, b e *ptr1: %d, %d, e %d\n", a, b, *ptr1);
*ptr1 = 50;
printf("a=%d, b=%d, *ptr1= %d", a, b, *ptr1);
```

b) (1.0V) Suponha que, depois do código da alínea a), são acrescentadas as três linhas seguintes:

```
ptr2 = &b;
*ptr2 = 100;
printf("O valor de a + b =%d", *ptr2);
```

Diga o que aparece no ecrã, após sua a execução.

7. (1.5V) Corrija/complete o código abaixo de forma a ordenar, por ordem decrescente, um vetor de inteiros

```
void ordenaVetor(int * vet, int N)
{
    for(i=2; i<N-1 && houveTroca; i++)
    {
        houveTroca=1;
        for (j=0; j<N-i-1; j++)
        {
            if (vet[j] > vet[j+1])
            {
                houveTroca=0;
                temp=vet[j];
                vet[j]=vet[j+1];
                vet[j+1]=temp;
            }
        }
    }
}
```