

Ficha de Apoio/Revisões/Preparação

1. Considere a função analise em baixo.

a) Dado um array de inteiros a, com 4 elementos, $a = [3 \ -2 \ 0 \ 4]$, determine o resultado da instrução: `printf("posicao=%d", analise(4, 0, a) + 1)`.

b) Explique o funcionamento da função analise, referindo os valores que devolve em consequência dos possíveis valores dos parâmetros n, chave, t.

```
int analise(int n, int chave, int *t) {  
    int i = 0;  
    while (t[i] != chave && i <= n-1)  
        i++;  
    if (i > n-1)  
        return -1;  
    else  
        return i;  
}
```

2. Desenvolva uma função que calcule a soma, a média e o máximo de um conjunto de números inteiros inseridos pelo utilizador, terminando a inserção, quando for especificado o valor 0. O programa deve **devolver** os valores calculados, bem como o número de valores inseridos.

3. Apresente um menu com 4 opções (para executar as funções f1(), f2(), f3() e f4()) + outra para sair, devendo aceitar sucessivamente opções e executando o código associado, até que o utilizador sinalize a intenção de terminar a execução.

4. Desenvolva uma função que calcule o somatório $\sum_{i=-1}^{n-1} i$, apresentando o resultado no ecrã. Faça o cálculo, de duas formas diferentes: usando ciclo *for* e outro ciclo.

5. Desenvolva uma função que calcule o somatório $\sum_{i=1}^{n+1} i * (i + 3)$, apresentando o resultado no ecrã. Faça o cálculo, de duas formas diferentes: usando ciclo *for* e outro ciclo.

6. Desenvolva uma função que apresente no ecrã os N primeiros números inteiros, por ordem decrescente.

7. Desenvolva uma função que apresente os primeiros N múltiplos de um número, exceto quando estes forem divisíveis por 2, 3 e 5.

8. Considere o seguinte código e apresente o que aparece no ecrã, se `val=4`.

```
#include<stdio.h>  
void main()  
{  
    int i, val, sum, prod;  
    printf("Insira um número: ");  
    scanf("%d", &val);  
    for (sum=0, i=prod=1; i<=val; i=i+1){  
        sum+=i;  
        prod*=i;}  
    printf("A Soma=%d; o Produto=%d\n\n", sum, prod);}
```

9. Desenvolva uma função que, dado um número inteiro positivo, calcule e devolva quanto falta para a próxima dezena.
10. Desenvolva uma função que, dado um número inteiro positivo, mostre os respetivos dígitos de forma invertida.
11. Desenvolva uma função que devolva quantas vezes surge um caracter na string, ambos passados como parâmetros.
12. Desenvolva uma função que devolva quantas vezes surge um caracter na string, ambos passados como parâmetros, mas ignorando se os caracteres são minúsculos ou maiúsculos.
13. Desenvolva uma função que permita copiar o conteúdo de uma string para outra.
14. Desenvolva uma função que inverta uma string.
15. Desenvolva uma função que coloque todos os primeiros caracteres de uma frase em maiúsculas, repetindo sucessivamente a funcionalidade, até que seja especificada uma string nula.
16. Desenvolva uma função que verifique se uma string é palíndromo (lê-se igualmente da esquerda para a direita e inversamente).
17. Crie uma função (strlen) que, dada uma string (um parâmetro da função), calcule e mostre o respetivo tamanho. O tamanho deve ser obtido, utilizando ponteiros para a string

18. Considere as estruturas de dados abaixo apresentadas:

```
#define MAX_V 20
```

```
typedef struct caract{  
    float percAlc; // % de alcool  
    short cor; // 1-tinto; 2-branco, 3-rosé; 4-  
    ruby;  
    char tpVinho: // M-mesa, G-generoso, E-  
    Espumante, C-Champanhe  
    char nome[50];  
}CARACT;  
  
typedef struct garrafa  
{  
    int ano;  
    CARACT def;  
    float preco;  
}GARRAFA;
```

a) Construa um vetor garrafeira com n garrafas, onde n é um valor dado pelo utilizador, mas que não deve ultrapassar uma constante MAX_V.

b) Desenvolva uma função que permita ler n garrafas e registá-las no vetor garrafeira.

```
void leGarr(GARRAFA v[ ], int n)
```

c) Desenvolva uma função que calcule e **devolva/disponibilize** os valores mínimo e máximo do preço das garrafas registadas no vetor.

```
void minMaxGarrafeira(GARRAFA v[ ], int n, int *max, int *min)
```

d) Desenvolva uma função que determine e devolva o preço e a marca da garrafa mais antiga. O nome desta função deverá ser

```
float precMarcaGVelha (GARRAFA v[ ], int n, string * antigo)
```

19. Pretende-se armazenar a informação relativa a um conjunto de DVDs de Filmes.

a) Defina uma estrutura de dados capaz de armazenar a informação relativa a cada DVD: título do filme, atores principais, produtor, ano, preço e tipo (A-Ação; E-Espionagem; F-Ficção Científica; R-Romance; P-Policial; T-Terror; W-Western; O-Outro).

b) Defina uma variável estruturada para armazenar a informação de um conjunto de DVDs, arquivados em 10 prateleiras com capacidade máxima de 100 DVDs.

c) Escreva uma função (usando ponteiros) de nome `nFilmePTipo`, que conte e devolva o número de filmes de cada tipo, entre anos, a especificar. Há um vetor que indica quantos filmes existem em cada prateleira e outro com os tipos dos filmes.

```
int * nFilmeTipo(DVD m[10][100], int anoIn, int anoFin)
```

20. Um grupo de estudantes de meteorologia tem de efetuar um estudo sobre as temperaturas médias registadas em várias cidades do nosso país durante os 12 meses de 2023. Considere que é considerado um único valor de temperatura (média) por cada mês e cada local, tal como consta na tabela.

Considere as seguintes definições.

```
#define N_CIDADES 50 //Número de cidades consideradas
#define N_MESES 12
#define MAX_CIDADE 30 //Número máximo de caracteres no nome da cidade
```

	jan	fev	mar	abr	mai	jun	jul	ago	set	out	nov	dez
Almada	13.25											
Braga												
...												
Viseu												

a) Defina um tipo ou tipos de dados que permitam armazenar a informação pretendida.

b) Defina a função/procedimento `máximoMes` que tomando os valores registados e um array auxiliar com `N_MESES` posições preencha este último com o valor máximo obtido, em todo o país, em cada mês.

c) Defina a função/procedimento que determine qual o mês e a cidade mais fria.

d) Suponha, agora, que satisfeitos com o programa, se pretende efetuar o registo não apenas para o ano de 2021, mas para os próximos 100 anos. Defina a estrutura de dados.

21. Considere a função aux. Explique o funcionamento da função quando se chama `aux(3,L,0,5)`, em que `L={-2,-2,1,3,5,6,7,8}`.

```
int aux(int x, int L[], int inic, int fim){
    int meio = (inic+fim)/2;
    if (inic > fim)
        return 0;
    else {
        if (x == L[meio])
            return 1;
        else{
            if (x > L[meio])
                return aux(x, L, meio+1, fim);
            else
                return aux(x, L, inic, meio-1);
        }
    }
}
```

22. Elabore uma função que calcule, de forma eficiente, o valor do somatório $\sum_{i=1}^N \frac{i!}{3^{i-1}}$

23. Considere a função mostra. Indique e justifique o que é mostrado no monitor caso a função seja invocada usando mostra(9,2).

```
void mostra(int n, int i){  
    if (n>1 && i>0){  
        mostra(n-i,i-1);  
    }  
    printf("\n%d",n+i);  
}
```