

# Programação 22/23 S2

Invalid Date

## 1 Informações gerais

### 1.1 Regras

- Podem consultar qualquer material da bibliografia recomendada.
- Podem consultar as folhas cujo conteúdo foi pré-aprovado.
- Não podem usar qualquer recurso que não esteja autorizado.
- Não podem usar qualquer tipo de comunicação com o exterior (telemóveis, computadores, etc.).
- Podem resolver o teste a lápis, excepto as questões de escolha múltipla ou verdadeiro/falso.
- A resolução do teste é individual.
- Não podem sair da sala durante o teste.
- **Nas respostas de escolha múltipla ou verdadeiro/falso, verifiquem se algumas das opções de resposta está na página seguinte.**

### 1.2 Cotação

- Componente teórica 40%
  - [8 valores] - Parte 1
    - \* Cada pergunta vale 1 valor
    - \* Respostas erradas descontam 25% do seu valor
      - Pergunta com 1 resposta, resposta errada desconta 0.25 valores
      - Pergunta com 2 respostas, cada resposta errada desconta 0.125 valores
      - etc.
  - [12 valores] - Parte 2
    - \* [3 valores] - Problema 1
    - \* [4 valores] - Problema 2
    - \* [5 valores] - Problema 3
- Componente prática 60%
- Nota final =  $0.4 * \text{componente teórica} + 0.6 * \text{componente prática}$

## 2 Componente teórica

### 2.1 Parte 1

#### 2.1.1 Pergunta 1

Quais das seguintes formas são válidas para criar um vetor com 10 `int` (2 respostas):

- a. `int vec[10];`
- b. `int vec = malloc(10*sizeof(int));`
- c. `int *vec = malloc(10);`
- d. `int *vec = malloc(10*sizeof(int));`
- e. `int vec[] = {0 * 10};`
- f. `int vec[] = {0} * 10;`

Solução: a, d

---

#### 2.1.2 Pergunta 2

Considera o seguinte programa:

```
int main() {  
    int array[10], soma=0;  
    for (int i=0; i<10; i++) *(array+i) = 2;  
    for (int i=0; i<10; i++) soma += array[i];  
    return 0;  
}
```

Escolhe a resposta correcta (1 resposta):

- a. O valor final de `soma` é 0.
- b. O valor final de `soma` é 10.
- c. Não é possível saber o valor de `soma` sem executar o programa.
- d. O programa não compila porque a variável `soma` não está inicializada.
- e. Nenhuma das respostas anteriores.

Solução: e

---

#### 2.1.3 Pergunta 3

Quais das seguintes formas são válidas ler um `int` do ficheiro binário `dados.bin` e guardar o valor lido na variável `x` (1 resposta):

```
int x;  
  
// opção A  
FILE *f = fopen("dados.bin", "r");  
fscanf(f, "%d", &x);  
  
// opção B  
FILE *f = fopen("dados.bin", "r");  
fread(&x, sizeof(int), 1, f);  
  
// opção C  
FILE *f = fopen("dados.bin", "rb");  
fread(&x, sizeof(int), 1, f);  
  
// opção D  
FILE *f = fopen("dados.bin", "ab");  
fread(&x, sizeof(int), 1, &f);
```

- a. Opção A
- b. Opção B
- c. Opção C
- d. Opção D
- e. Nenhuma das respostas anteriores.

Solução: c

---

#### 2.1.4 Pergunta 4

No final da execução do seguinte código, qual é o valor da variável x? (1 resposta)

```
void funcao1(int * arg) {
    *arg = 1;
}

void funcao42(int arg) {
    arg = 42;
}

int main() {
    int x = 2;
    int *y = &x;

    funcao1(y);
    funcao42(*y);

    return 0;
}
```

- a. 0
- b. 1
- c. 2
- d. 42
- e. Nenhuma das respostas anteriores.

Solução: b

---

#### 2.1.5 Pergunta 5

Qual das seguintes estruturas é adequada para uma lista duplamente ligada: (1 resposta)

- a. struct node { int value; struct node \*next; };
- b. struct node { int value; struct node \*next; struct node \*prev; };
- c. struct node { int value; struct node \*prev; };
- d. struct node { int value; struct node \*next; struct node \*prev; struct node \*next; };
- e. Nenhuma das respostas anteriores.

Solução: b

---

#### 2.1.6 Pergunta 6

Considera o seguinte programa:

```
for (int i = 0; i < 10; i++) {
    if (i % 2 == 0) printf("i=%d\n", i);
}

for (int i = 0; i < i+10; i+=2) {
    printf("i=%d\n", i);
}
```

Escolhe a resposta correcta (1 resposta):

- Os dois ciclos fazem exactamente o mesmo.
- O primeiro ciclo imprime os números ímpares de 0 a 9, enquanto o segundo imprime os números pares de 0 a 10.
- O primeiro ciclo imprime os números pares de 0 a 10, enquanto o segundo imprime os números ímpares de 0 a 9.
- O primeiro ciclo é um ciclo infinito, enquanto o segundo imprime os números pares de 0 a 10.
- O segundo ciclo é um ciclo infinito, enquanto o primeiro imprime os números pares de 0 a 10.
- Nenhuma das respostas anteriores.

Solução: e

### 2.1.7 Pergunta 7

Se quisermos escrever uma função que devolve um vetor de inteiros, qual das seguintes opções é a correcta? (1 resposta)

- `int * funcao() { int vec[10]; return vec; }`
- `int * funcao() { int *vec = malloc(10*sizeof(int)); return vec; }`
- `int * funcao() { int vec[10]; return &vec; }`
- `int * funcao() { int *vec = malloc(10*sizeof(int)); return &vec; }`
- Nenhuma das respostas anteriores.

Solução: b

### 2.1.8 Pergunta 8 - switch

Considera o seguinte programa:

```
int main() {
    int x = 5;
    switch (x) {
        case 1:
            printf("1\n");
        case 2:
            printf("2\n");
        case 3:
            printf("3\n");
        default:
            printf("0\n");
    }
    return 0;
}
```

Escolhe a resposta correcta (1 resposta):

- O programa imprime 0.
- O programa imprime 1.
- O programa imprime 2.
- O programa imprime 3.
- O programa imprime 0, 1, 2 e 3.
- Nenhuma das respostas anteriores.

Solução: a

## 2.2 Parte 2

### 2.2.1 Problema 1 [3 valores]

- Considera o seguinte código.

```
#include <stdio.h>
#include <string.h>

#define MAX_LINHAS 100
#define MAX_COLUNAS 100

typedef struct {
    int n_linhas, n_colunas;
    float vals[MAX_LINHAS][MAX_COLUNAS];
} Matriz;

Matriz operation(Matriz m1) {
    Matriz m2;
    m2.n_linhas = m1.n_colunas;
    m2.n_colunas = m1.n_linhas;
    for (int i = 0; i < m1.n_linhas; i++) {
        for (int j = 0; j < m1.n_colunas; j++) {
            m2.vals[j][i] = m1.vals[i][j];
        }
    }
    return m2;
}

void print_matrix(Matriz *m) {
    for (int i = 0; i < m->n_linhas; i++) {
        for (int j = 0; j < m->n_colunas; j++) {
            printf("%f ", m->vals[i][j]);
        }
        printf("\n");
    }
}

int main() {
    Matriz m1 = {2, 3, {{1, 2, 3}, {4, 5, 6}}};
    Matriz m2 = operation(m1);
    print_matrix(&m2);
    return 0;
}
```

- [1.5 val] Indica o output do programa na consola.
- [1 val] Explica o que o programa está a fazer.
- Qual é o tamanho máximo que uma matriz pode ter?

Solução a. o output do programa é:

```
```text
1.000000 4.000000
2.000000 5.000000
3.000000 6.000000
```
```

- O programa está a fazer a transposta de uma matriz. A função `operation` recebe uma matriz `m1` e devolve uma nova matriz `m2` com a transposta de `m1`. A função `print_matrix` imprime uma matriz no ecrã.
- O tamanho máximo que uma matriz pode ter é `MAX_LINHAS` linhas e `MAX_COLUNAS` colunas, que neste caso é 100 linhas e 100 colunas.

### 2.2.2 Problema 2 - ordenar índices [4 valores]

- Escreve uma função que recebe um array de reais e o seu tamanho.
- A função cria um novo vetor de inteiros positivos inicializados com os índices do array de reais.
- A função devolve este vetor.
- A função ordena os índices de forma a que o array de reais fique ordenado de forma decrescente.
- Exemplo:
  - No início
    - \* vetor de reais (1.2, 3.4, 2.1, 0.5)
    - \* vetor dos índices (0, 1, 2, 3), depois de criado e inicializado
  - No final
    - \* vetor de reais (1.2, 3.4, 2.1, 0.5), i.e. não é alterado
    - \* vetor dos índices (1, 2, 0, 3), i.e. índices ordenam o vetor de reais de forma decrescente
  - Valor de retorno é o vetor de índices ordenado (1, 2, 0, 3)
- Solução

```
// troca valores
void swap(int * a, int * b) {
    int tmp = *a;
    *a = *b;
    *b = tmp;
}

int * ordenar_indices(float * array, int n) {
    // criar vetor de indices
    int * indices = malloc(n * sizeof(int));
    // inicializar indices
    for (int i = 0; i < n; i++) indices[i] = i;
    // ordenar indices
    for (int i = 0; i < n; i++) {
        for (int j = i+1; j < n; j++) {
            // se o valor do array no indice i for menor que o valor do array no indice j
            if (array[indices[i]] < array[indices[j]]) {
                swap(&indices[i], &indices[j]);
            }
        }
    }
    return indices;
}
```

### 2.2.3 Problema 3 - evolução de depósito a prazo e relatório [5 valores]

- Escreva um programa completo que calcula a evolução de um depósito a prazo.
- O programa deve ler do teclado o valor inicial do depósito, a taxa de juro anual, o ano inicial e o último ano.
- O programa deve calcular o valor do depósito para cada ano e escrever o resultado num ficheiro de texto.
- O nome do ficheiro de texto também é recebido do utilizador.
- Todos os dados recebidos do utilizador são pedidos na main.
- A lógica do programa deve ser toda implementada numa função `evolucao_deposito` que recebe os inputs e escreve o output no formato indicado no ficheiro indicado
- Exemplo:

- Input a pedir ao utilizador (os valores depois dos : são os valores introduzidos pelo utilizador, que podem ser diferentes dos do exemplo)

```
Insira os seguintes dados
Valor inicial do depósito: 1000
Taxa de juro anual: 5
Ano inicial: 2010
Ano final: 2015
Nome do ficheiro de output: deposito.txt
```

- Output, ficheiro `deposito.txt` com o seguinte conteúdo:

```
2010 1000.00
2011 1050.00
2012 1102.50
2013 1157.62
2014 1215.51
2015 1276.28
```

- Solução

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void evolucao_deposito(float valor_inicial, float taxa_juro, int ano_inicial, int ano_final, char *nome_ficheiro) {
    FILE *f = fopen(nome_ficheiro, "w");
    float valor = valor_inicial;
    for (int ano = ano_inicial; ano <= ano_final; ano++) {
        fprintf(f, "%d %.2f\n", ano, valor);
        valor = valor * (1 + taxa_juro/100);
    }
    fclose(f);
}

int main() {
    float valor_inicial, taxa_juro;
    int ano_inicial, ano_final;
    char nome_ficheiro[100];
    printf("Valor inicial: ");
    scanf("%f", &valor_inicial);
    printf("Taxa de juro anual: ");
    scanf("%f", &taxa_juro);
    printf("Primeiro ano: ");
    scanf("%d", &ano_inicial);
    printf("Último ano: ");
```

```
scanf("%d", &ano_final);  
printf("Nome do ficheiro: ");  
scanf("%s", nome_ficheiro);  
evolucao_deposito(valor_inicial, taxa_juro, ano_inicial, ano_final, nome_ficheiro);  
return 0;  
}
```

### 3 Componente prática

Se tiver obtido aprovação na componente prática (laboratórios e projeto), a nota desta componente é a obtida anteriormente. Em caso de reprovação, ou melhoria de nota, à componente prática, pode optar por substituir essa nota pela obtida em exame oral.