

PROCESSO SELETIVO - 06/2025

Área de Conhecimento: Algoritmos e Linguagem de Programação

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 1

Em programas que utilizam grande quantidade de memória, a alocação deste recurso deve ser realizada com cuidado. Em algumas circunstâncias, o uso da memória pode ser otimizado com a utilização de registros variantes. Em linguagens de programação como C, o registro variante é construído através de uma união disjuntiva. Considerando o código a seguir, que utiliza a declaração do tipo Ponto2D em linguagem C, (2,0 pontos)

```
#include <stdio.h>

typedef union t_Ponto2D {
    int data[2];
    struct {
        int X, Y;
    };
} Ponto2D;

int main() {
    Ponto2D p1;
    p1.data[0] = 1;
    p1.data[1] = 2;
    p1.X = 3;
    p1.Y = 4;
    printf ("%d %d %d %d\n", p1.data[0], p1.data[1], p1.X, p1.Y);
    return 0;
}
```

A saída em tela do código será:

- a) 1 2 1 2
- b) 1 2 3 4
- c) 3 4 1 2
- d) 3 4 3 4
- e) 4 3 2 1

d

PROCESSO SELETIVO - 06/2025

Área de Conhecimento: Algoritmos e Linguagem de Programação

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 2

Na teoria de sistemas, define-se elemento minimax de uma matriz como o menor elemento da linha em que se encontra o maior elemento da matriz. Escreva um programa de computador em linguagem C que leia os valores de uma matriz A de tamanho NxM (onde N e M são valores pré-definidos e constantes), e que determine o elemento minimax desta matriz, escrevendo na tela o seu valor e suas coordenadas (linha, coluna). É permitido, além das próprias, o uso apenas de funções/procedimentos de biblioteca padrão de entrada e saída de dados. (2,0 pontos)

```
#include <stdio.h>

#define N 3    // número de linhas
#define M 4    // número de colunas

int main() {
    int A[N][M];
    int i, j;

    // Leitura da matriz
    printf("Digite os elementos da matriz %dx%d:\n", N, M);
    for (i = 0; i < N; i++) {
        for (j = 0; j < M; j++) {
            scanf("%d", &A[i][j]);
        }
    }

    // 1. Encontrar o maior elemento da matriz e sua linha
    int maior = A[0][0];
    int linha_maior = 0;

    for (i = 0; i < N; i++) {
        for (j = 0; j < M; j++) {
            if (A[i][j] > maior) {
                maior = A[i][j];
                linha_maior = i;
            }
        }
    }

    // 2. Encontrar o menor elemento da linha e sua coluna
    int menor = A[linha_maior][0];
    int coluna_menor = 0;

    for (j = 0; j < M; j++) {
        if (A[linha_maior][j] < menor) {
            menor = A[linha_maior][j];
            coluna_menor = j;
        }
    }

    // Saída da matriz
    printf("Matriz:\n");
    for (i = 0; i < N; i++) {
        for (j = 0; j < M; j++) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }

    // Saída do resultado
    printf("Elemento minimax: %d (%d, %d)\n", menor, linha_maior + 1, coluna_menor + 1);
}
```

```

        }
    }

// 2. Encontrar o menor elemento da linha do maior → MINIMAX
int minimax = A[linha_maior][0];
int coluna_minimax = 0;

for (j = 1; j < M; j++) {
    if (A[linha_maior][j] < minimax) {
        minimax = A[linha_maior][j];
        coluna_minimax = j;
    }
}

// 3. Saída
//printf("\nMaior elemento da matriz: %d (linha %d)\n", maior,
linha_maior);
printf("Elemento minimax: %d\n", minimax);
printf("Coordenadas do minimax: linha %d, coluna %d\n",
    linha_maior, coluna_minimax);

return 0;
}

```

PROCESSO SELETIVO - 06/2025

Área de Conhecimento: Algoritmos e Linguagem de Programação

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 3

Arquivo é um mecanismo de abstração que permite a manipulação de dados de maneira persistente, concorrente e em grandes quantidades. Sobre o assunto, considere as afirmativas a seguir.

- I. O conteúdo de arquivo é estritamente binário, ou seja, não é possível armazenar informações textuais. Por isso, quando existe a necessidade de armazenar um arquivo texto, do ponto de vista do programador, é necessário converter cada caractere em um número inteiro (tipo int) antes de gravar o texto desejado.
- II. Uma maneira comum de armazenar dados em arquivos é através de sequência de bytes não estruturados. Independente da maneira como os dados são armazenados, a interpretação do significado desses dados deve ser feita pelos programas de usuário.
- III. Todo sistema operacional armazena um certo conjunto de informações junto a cada arquivo, conhecidas como atributos ou metadados. Dentre as informações armazenadas pelos metadados de um arquivo em um sistema, podem estar: identificador do arquivo; hora da criação; último acesso; última mudança; visibilidade; tipo de arquivo.
- IV. Para usar um arquivo em C é necessário abri-lo. Para tanto, a linguagem C possui o comando fopen. Exemplo: **arq = fopen("ArqGrav.txt", "wt");** Neste comando de exemplo, o argumento de modo "wt" (write text) informa a criação de um novo arquivo de texto no disco e o abre para gravação.

Assinale a alternativa correta. **(1,5 pontos):**

- a) Somente as afirmativas I e II são corretas.
- b) Somente as afirmativas I e IV são corretas.
- c) Somente as afirmativas II e III são corretas.
- d) Somente as afirmativas II, III e IV são corretas.
- e) Todas estão corretas

d) Somente as afirmativas II, III e IV são corretas.

PROCESSO SELETIVO - 06/2025

Área de Conhecimento: Algoritmos e Linguagem de Programação

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 4

Escreva um programa de computador, em linguagem de programação C, que leia a data de início de uma tarefa (dia inicial e mês inicial) e a data de término da mesma (dia final e mês final). O programa deve calcular o número de dias que se passaram entre as duas datas. Considerações: **(2,0 pontos)**

- A data final é posterior à data inicial;
- Ambas as datas estão dentro do mesmo ano;
- O ano não é bissexto;
- Os meses têm diferentes números de dias;
- Não é necessário realizar validação dos dados de entrada.
- É permitido, além das próprias, o uso apenas de funções/procedimentos de biblioteca padrão de entrada e saída de dados.

```
#include <stdio.h>

int main() {
    int dia_i, mes_i, dia_f, mes_f;
    int dias_mes[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int dias = 0;

    // Entrada
    printf("Digite o dia inicial: ");
    scanf("%d", &dia_i);
    printf("Digite o mes inicial: ");
    scanf("%d", &mes_i);

    printf("Digite o dia final: ");
    scanf("%d", &dia_f);
    printf("Digite o mes final: ");
    scanf("%d", &mes_f);

    // Se as datas estão no mesmo mês
    if (mes_i == mes_f) {
```

```
dias = dia_f - dia_i;
} else {
    // Dias restantes do mês inicial
    dias += dias_mes[mes_i - 1] - dia_i;

    // Soma dias dos meses completos entre as datas
    for (int m = mes_i + 1; m < mes_f; m++) {
        dias += dias_mes[m - 1];
    }

    // Adiciona dias do mês final
    dias += dia_f;
}

// Saída
printf("Dias decorridos: %d\n", dias);

return 0;
}
```

PROCESSO SELETIVO - 06/2025

Área de Conhecimento: Algoritmos e Linguagem de Programação

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 5:

Sobre recursividade, Wirth (1989) indica que “a ferramenta necessária para se codificar programas de modo recursivo é o procedimento ou sub-rotina, que permite que seja dado um nome a um comando, comando esse que pode chamar a si próprio através deste nome”. Sendo assim, escreva um programa de computador que por meio de uma função recursiva, em linguagem C, calcule o valor da série **S** descrita a seguir para um valor inteiro $n > 0$, a ser fornecido como parâmetro para a mesma. A função deve retornar o valor de **S**. Para o cálculo do fatorial também desenvolva funções, uma iterativa e outra recursiva. É permitido, além das próprias, o uso apenas de funções/procedimentos de biblioteca padrão de entrada e saída de dados. (2,5 pontos).

$$S = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$$

```
#include <stdio.h>

// opcao 1 - fatorial com versão recursiva
int fatorial_rec(int valor) {
    if(valor == 0)
        return 1;
    else {
        return (valor * fatorial_rec(valor-1));
    }
}

// opcao 2 - fatorial com versão iterativa
int fatorial_it(int valor) {
    int i, fat=1;
    for(i=1;i<=valor;i++){
        fat *= i;
    }
    return fat;
}

// somatorio com versão recursiva (obrigatoriamente)
float somatorio_rec(int valor) {
    printf("\n\n====>> n = %d\n", valor);

    if(valor == 1)
```

```
    return 1;
else {
    return (1.0/fatorial_rec(valor-1) + somatorio_rec(valor-1));
    // chamada à função fatorial poderia ser para a versão iterativa do
fatorial
}
}

void main(){
    int n;
    printf("Digite o valor de n para a série S: ");
    scanf("%d", &n);
    printf("\nO valor da série com n = %d é: %f\n",n, somatorio_rec(n));
}
```

Membros da Banca:

[assinado digitalmente]
Gilmario Barbosa dos Santos

[assinado digitalmente]
André Tavares da Silva

[assinado digitalmente]
Presidente da Banca: Adriano Fiorese



Código para verificação: **E78I2W2G**

Este documento foi assinado digitalmente pelos seguintes signatários nas datas indicadas:

 **ADRIANO FIORESE** (CPF: 743.XXX.339-XX) em 24/11/2025 às 10:55:46

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:36:39 e válido até 30/03/2118 - 12:36:39.
(Assinatura do sistema)

 **ANDRE TAVARES DA SILVA** (CPF: 908.XXX.020-XX) em 24/11/2025 às 13:29:22

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:44:17 e válido até 30/03/2118 - 12:44:17.
(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link <https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTIwMjJfMDAwNDY0NTNfNDY0ODRfMjAyNV9FNzhJMlcyRw==> ou o site <https://portal.sgpe.sea.sc.gov.br/portal-externo> e informe o processo **UDESC 00046453/2025** e o código **E78I2W2G** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.