PROCESSO SELETIVO 03/2025

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 1:

Sobre as vantagens de criar funções em programação, segundo Forbellone & Eberspächer (2000) afiam que "quando modularizamos um algoritmo, procuramos identificar problemas de instância menor que possam ser desenvolvidos separadamente, pois possuem uma independência relativa ao conjunto solução total". Considerando o uso de funções na linguagem C, desenvolva um programa nesta linguagem que permita calcular o fatorial de um conjunto de números compreendido entre **n1** e **n2** (inclusive). Deve-se desenvolver a validação da entrada, no sentido de garantir que **n2** > **n1**, bem como observar que o fatorial é calculado apenas para números inteiros positivos. O cálculo de cada fatorial deve ser executado através de uma função. A função deve receber um valor inteiro e retornar o fatorial deste valor. A entrada de dados bem como a impressão dos fatoriais deve ser feita na função principal.

```
#include <stdio.h>

int fatorial(int valor) {
    int resultado = 1;
    if(valor == 0)
        return 1;
    else {
        return (valor * fatorial(valor-1));
    }
}

int main() {
    int i, j, n1, n2, condicao = 0, maximo = 100, neg = 0;
    int vetor_valores[maximo], fat, total, resultado;

    while(condicao == 0) {
        printf("Digite o valor de n1 (n1 deve ser menor que n2):\n");
        scanf("%d", &n1);
        printf("Digite o valor de n2 (n2 deve ser maior que n1):\n");
        scanf("%d", &n2);
        if(n2>n1) {
            condicao = 1;
            printf("\nEntrada validada!");
        }
        else {
            printf("\nEntrada invalidada!\n\n");
        }
    }

    // RESPOSTA CONTINUA NA PRÓXIMA FOLHA
```

Pág. 01 de 09 - Documento assinado digitalmente. Para conferência, acesse o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo UDESC 00024120/2025 e o código U0V71OJ4.

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

```
    printf("\n\nn1 = %d, n2 = %d\n", n1, n2);
    total = n2 - n1 + 1;

    j=0;
    for (i = n1; i <= n2; i++) {
        if (i>=0) {
            resultado = fatorial(i);
            vetor_valores[j] = resultado;
            j = j + 1;
            printf(" %d", i);
            printf(" %d\n", vetor_valores[j-1]);
        }
        else {
            neg = neg + 1;
            printf("%d", i);
            printf(" x\n");
        }
    }

    printf("\n>> ");
    for (i = 0; i <= (n2 - n1 - neg); i++)
        printf("%d ", vetor_valores[i]);
    printf("\n\n");

    return 0;
}
```

QUESTÃO 2:

De acordo com Manzano & Oliveira (2014), função é um bloco de código de programa que contém início e fim, sendo identificada por um nome de referência. “É uma estrutura de abstração de dados que possui como característica operacional a capacidade de sempre retornar um conteúdo como resposta de sua ação. No entanto, funções também operam com passagens de parâmetro por valor e por referência”. Assim, existem dois mecanismos para implementação de passagem de parâmetros. O mecanismo de passagem por valor envolve a criação de uma cópia do parâmetro real no ambiente local do subprograma (função). O mecanismo de passagem por referência envolve a criação de uma referência (endereço) ao parâmetro real no ambiente local do subprograma (função). Escreva um programa na linguagem de C que exemplifique claramente a passagem de parâmetros por valor bem como por referência. Faça comentários no seu código para esclarecer seu exemplo.

Pág. 02 de 09 - Documento assinado digitalmente. Para conferência, acesse o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo UDESC 00024120/2025 e o código U0V71OJ4.

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

```
// O exemplo usa a função de fatorial
// Os candidatos podem apresentar outros exemplos desde que sejam coerentes
// com os conceitos solicitados

// Exemplo 1 - passagem de parâmetro por valor
// Neste exemplo, variável v recebe a cópia do valor da variável x
int fatorial_iterativo(int v) {
     int i, fat=1;
     for(i=1;i<=v;i++){
        fat *= i;
    }
    return fat;
}

int main(){
   int x=5;
   int f;

   f=fatorial(x);

   printf("f=%d - x=%d",f);

   return 0;
}

// Exemplo 2 - passagem de parâmetro por referência
// Neste exemplo, variável fat recebe o endereço (referência) da variável f
// A função foi construída de modo que não necessita de retorno pois a passagem
// por referência irá alterar o valor da variável f da função principal
void fatorial_iterativo(int v, int *fat) {
    int i;
    *fat = 1;
    for(i=1;i<=v;i++){
        *fat *= i;
    }
}

int main(){
   int x=5;
   int f;

   fatorial(x,&f);

   printf("f=%d - x=%d",f);

   return 0;
}
```

Pág. 03 de 09 - Documento assinado digitalmente. Para conferência, acesse o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo UDESC 00024120/2025 e o código U0V71OJ4.

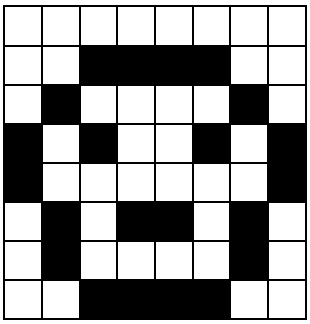
PROCESSO SELETIVO 03/2025

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 3:

Segundo Souza et al. (2005), um dos algoritmos mais fáceis para se compactar uma imagem é aquele conhecido como RLE (*Run Length Encoding*) pois percorre um arquivo tipo *bitmap* e o descreve com o número de vezes que o mesmo tipo de *bit* se repete na imagem. Como exemplo, os autores apresentam uma matriz de 8x8 representando uma imagem em preto e branco que pode ser igualmente representada como uma matriz de valores inteiros, com o valor 1 indicando os pixels pretos e o valor 0 representando os pixels brancos.



0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
1	0	1	0	0	1	0	1
1	0	0	0	0	0	0	1
0	1	0	1	1	0	1	0
0	1	0	0	0	0	1	0
0	0	1	1	1	1	0	0

A proposta de representação compactada (RLE) poderia ser feita usando a letra '**B**' para indicar pixels brancos e a letra '**P**' para indicar pixels pretos, com um número à esquerda indicando a quantidade de pixels subsequentes de um mesmo tipo. Assim, '**8B**' indicaria 8 pixels brancos e '**2B4P**' indicaria 2 pixels brancos seguidos de 4 pixels pretos. Adicionalmente, poderia-se usar um '**0**' (zero) para indicar um final de linha e dois '**00**' (zeros) para indicar o fim da imagem. Com isso, a imagem *bitmap* acima poderia ser representada como uma *string* (sequência de caracteres) da seguinte forma:

"8B02B4P2B01B1P4B1P1B01P1B1P2B1P1B1P01P6B1P01B1P1B2P1B1P1B01B1P4B1P1B02B4P2B00"

Com base nesta mesma proposta de representação de imagem através de *string*, escreva um programa em linguagem C que leia (via teclado) uma matriz de 8x8 composta de valores inteiros (zeros e uns) representando uma imagem *bitmap* e crie uma *string* com a representação da imagem usando o mesmo padrão proposto acima (números, letras B e P e valores zero). A *string* criada deve ser apresentada na tela ao final. Dica: Na linguagem C, para converter um dígito inteiro para um caracter, pode-se armazenar seu valor acrescido de 48 (valor correspondente da tabela ASCII). Exemplo: **char c=1+48;** // armazena o dígito '**1**' como um caracter na variável c.

Pág. 04 de 09 - Documento assinado digitalmente. Para conferência, acesse o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo UDESC 00024120/2025 e o código U0V71OJ4.

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

```
#include <stdio.h>
#include <string.h>
#define L 8  // opcional
#define C 8  // opcional

int main() {
    int i, j;
    int mat[L][C];
    char padrao[150];  // string para armazenar padrão do bitmap.
                       // Como exemplo, adotou-se o limite de 150 caracteres.

     // leitura dos valores da matriz
     for(i=0;i<L;i++){
          for(j=0;j<C;j++){
               scanf("%d",&mat[i][j]);
          }
     }

     int contB=0, contP=0; // contagem de pixels brancos e pixels pretos
     int c=0;  // contagem de caracteres da string
     for(i=0;i<L;i++){
          for(j=0;j<C;j++){
               if(j>0){
                    if(mat[i][j]!=mat[i][j-1]){
                         if (contB>0){
                              padrao[c++]=48+contB;  // quant. de pixels brancos
                              padrao[c++]='B';       // armazena a letra B
                              contB=0;
                         }
                         if (contP>0){
                              padrao[c++]=48+contP;  // quant. de pixels pretos
                              padrao[c++]='P';       // armazena a letra P
                              contP=0;
                         }
                    }
               }
               if(mat[i][j]==1)
                    contP++;      // aumenta a contagem de pixels pretos
               if(mat[i][j]==0)
                    contB++;      // aumenta a contagem de pixels brancos
          }
          if (contB>0){
               padrao[c++]=48+contB;    // quant. de pixels brancos
               padrao[c++]='B';         // armazena a letra B
               contB=0;
          }

          // CONTINUA NA PRÓXIMA FOLHA
```

Pág. 05 de 09 - Documento assinado digitalmente. Para conferência, acesse o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo UDESC 00024120/2025 e o código U0V71OJ4.

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

```
        if (contP>0){
              padrao[c++]=48+contP;    // quant. de pixels pretos
              padrao[c++]='P';         // armazena a letra P
              contP=0;
        }
        padrao[c++]='0';     // armazena dígito zero, indicando fim de linha
    }
    if (contB>0){
        padrao[c++]=48+contB;       // quant. de pixels brancos
        padrao[c++]='B';            // armazena a letra B
    }
    if (contP>0){
        padrao[c++]=48+contP;       // quant. de pixels pretos
        padrao[c++]='P';            // armazena a letra P
    }
    padrao[c++]='0';
    padrao[c++]='0';          // dois dígitos zeros = fim da imagem
    padrao[c++]='\0';         // '\0' como finalizador da string

    printf("\n|%s|\n",padrao);   // mostra na tela a string criada do bitmap

    return 0;
}
```

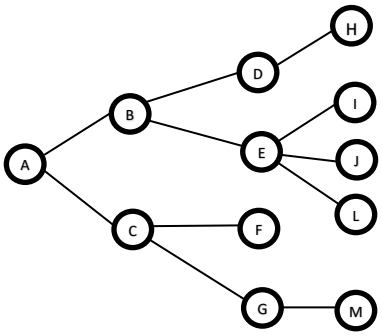
QUESTÃO 4:

Segundo Forbellone & Eberspächer (2000), uma árvore é uma lista na qual cada elemento possui dois ou mais sucessores, porém todos os elementos possuem apenas um antecessor. Um exemplo de árvore é apresentado na figura abaixo. Neste exemplo, os nós são representados por círculos. O nó 'A', que dá origem aos demais, é chamado de nó **raiz**. Os nós sucessores de outro nó são chamados **filhos** e o antecessor de cada nó é chamado de **pai**. Cada elemento final da árvore (sem sucessores) é chamado de nó **folha**. No exemplo, o nó A é o nó raiz. Os nós B e C são filhos do nó A, assim como I, J e L são nós filhos do nó E (entre outros casos). O nó B é pai dos nós D e E (entre outros casos). Os nós H, I, J, L, F e M são nós folhas pois não possuem descendentes.

Pág. 06 de 09 - Documento assinado digitalmente. Para conferência, acesse o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo UDESC 00024120/2025 e o código U0V71OJ4.

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA



Considerando uma estrutura de árvore na linguagem C definida como a seguir, construa uma função que permita excluir nós-folha desta árvore. Considere uma estrutura de **árvore binária**.

```
struct noh {
      int info;
      struct noh *esquerda;
      struct noh *direita;
};
typedef struct noh *PTR_NOH;
```

```
PTR_NOH remove_folhas( PTR_NOH n ){
     if( n == NULL )
          return NULL;

     if( n->esquerda == NULL && n->direita == NULL ){ // Este é um nó folha.
          free( n );
          return NULL;
     }
     else{
          n->esquerda = remove_folhas( n->esquerda );
          n->direita = remove_folhas( n->direita );
          return n;
     }
}
```

Pág. 07 de 09 - Documento assinado digitalmente. Para conferência, acesse o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo UDESC 00024120/2025 e o código U0V71OJ4.

PROCESSO SELETIVO 03/2025

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

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, faça uma função recursiva, em linguagem C, que 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**.

S = 1 + 1/1! + 1/2! + 1/3! + ... + 1/n!

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

// opcao 2 - fatorial com versão iterativa – não obrigatório
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 == 0)
        return 1;
    else {
        return ( 1.0/fatorial_rec(valor) + somatorio_rec(valor-1));
        // chamada à função fatorial poderia ser para a versão iterativa do fatorial
    }
}
```

Pág. 08 de 09 - Documento assinado digitalmente. Para conferência, acesse o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo UDESC 00024120/2025 e o código U0V71OJ4.

PROCESSO SELETIVO 03/2025

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 6:

Tenenbaum, Langsam & Augenstein (1995) afirmam que "C permite que o programador referencie a posição de objetos bem como os próprios objetos (isto é, o conteúdo de suas posições). (...) É possível declarar uma variável cujo tipo de dado seja um ponteiro e cujos possíveis valores sejam posições de memória". Considerando os conceitos relativos a ponteiros em C, qual será a saída (em tela) deste programa supondo que **i** ocupa o endereço **4094** na memória?

```
int main() {
        int i=5, *p;
        p = &i;
        printf("%d %d %d %d %d \n", p, *p+2, **&p, 3*(*p), *&p);
        return 0;
}
```

```
// A saída esperada para o programa é:

4094 7 5 15 4094

// Comentários sobre as respostas (não fazem parte da resposta obrigatória):
// 1° valor – 4094 = conteúdo do ponteiro p (endereço de i)
// 2° valor – 7    = conteúdo apontado pelo ponteiro p acrescido de 2 (i+2=5+2)
// 3° valor – 5    = conteúdo do conteúdo do endereço do ponteiro p (valor de i)
// 4° valor – 15   = conteúdo apontado pelo ponteiro p vezes 3 (i*3=5*3)
// 5° valor – 4094 = conteúdo do endereço do ponteiro p (endereço de i)
```

*O padrão de resposta deve estar fundamentado nas bibliografias exigidas pelo Edital, para evitar problemas o professor deverá citar o capítulo/pagína do livro utilizado.

Membros da Banca:

(assinado digitalmente)

Avaliador 1: Charles Christian Miers

(assinado digitalmente)

Avaliador 2: Ricardo Ferreira Martins

(assinado digitalmente)

Presidente da Banca: Luciana Rita Guedes

Pág. 09 de 09 - Documento assinado digitalmente. Para conferência, acesse o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo UDESC 00024120/2025 e o código U0V71OJ4.

Assinaturas do documento

Código para verificação: **U0V71OJ4**

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

LUCIANA RITA GUEDES (CPF: 684.XXX.129-XX) em 07/07/2025 às 13:07:34
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:48:12 e válido até 30/03/2118 - 12:48:12.
(Assinatura do sistema)

CHARLES CHRISTIAN MIERS (CPF: 847.XXX.229-XX) em 07/07/2025 às 13:27:51
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:34:33 e válido até 30/03/2118 - 12:34:33.
(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTIwMjJfMDAwMjQxMjBfMjQxMzdfMjAyNV9VMFY3MU9KNA== ou o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo **UDESC 00024120/2025** e o código **U0V71OJ4** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.