

PROCESSO SELETIVO - 03/2021

Área de Conhecimento: Ciência da Computação/Algoritmo e Estruturas de Dados

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 1: 1(um) ponto

```
#include <stdlib.h>
#include <stdio.h>
int main( )
{ float a,b,c;

printf("Digite um valor para o lado A:\n");
scanf("%f", &a);
printf("Digite um valor para o lado B:\n");
scanf("%f", &b);
printf("Digite um valor para o lado C:\n");
scanf("%f", &c);
if ((a+b>c)&&(a+c>b)&&(c+b>a))
{
    if (a==b && b==c)
        printf("\n\nO triangulo eh equilatero.\n\n");
    else if (a==b || a==c || b==c)
        printf("\n\nO triangulo eh isoceles.\n\n");
    else
        printf("\n\nO triangulo eh escaleno.\n\n");
}
else
    printf("\n\nNao ha triangulo!!\n\n");
}
```

FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico, Lógica de Programação. Makron Books, São Paulo, 2000. ISBN 857194718X - Capitulo 2 , paginas 39 e 40

Membros da Banca:

Prof. Carlos Norberto Vetorazzi Junior (nome e assinatura)

Prof. Ricardo Ferreira Martins (nome e assinatura)

**Profa. Éverlin Fighera Costa Marques
Presidente da Banca (nome e assinatura)**

PROCESSO SELETIVO - 03/2021

Área de Conhecimento: Ciência da Computação/Algoritmo e Estruturas de Dados

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 2: 1.5 ponto

```
#include <stdio.h>
int fatorial_i(int n) // funcao iterativa
{
    int i, f;
    f=1;
    for(i=2;i<=n;i++)
        f=f*i;
    return(f);
}
int fatorial_r(int n) // funcao recursiva
{
    int i,f;
    if (n==0) return(1);
    else return(n*fatorial_r(n-1));
}
int main()
{
    int n, fat;
    printf("Digite um numero inteiro positivo: ");
    scanf("%d", &n);
    fat = fatorial_i(n);
    printf("Funcao iterativa: %d\n", fat);
    fat = fatorial_r(n);
    printf("Funcao recursiva: %d\n", fat);
}
```

GOODRICH, Michael T; TAMASSIA, Roberto; COPSTEIN, Bernardo. Projeto de algoritmos: fundamentos, análise e exemplos da internet. Porto Alegre: Bokman, 2004. Capítulo 1
TENENBAUM, A.M.; LANGSAM, Y.; AUGENSTEIN, M.J. Estruturas de Dados Usando C. São Paulo: Makron Books, 1995. ISBN 85-346-0348-0 - Capítulo 3

Membros da Banca:

Prof. Carlos Norberto Vetorazzi Junior

Prof. Ricardo Ferreira Martins

Profa. Éverlin Fighera Costa Marques

PROCESSO SELETIVO - 03/2021

Área de Conhecimento: Ciência da Computação/Algoritmo e Estruturas de Dados

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 3: __ 1.5 ponto _____

```
#include <stdio.h>
#include <stdlib.h>
int verifica_primo(int n) {
int count, i;
count = 0;
for (i=1; i<=n; i++) {
if((n % i) == 0)
count = count+1;
}
if(((count==2) || (n==1)) && (n!=1))
return 1;
else
return 0;
}
int main(void) {
int i, n, j, r, tot, tot_num_linha;
printf("\nDigite o valor de N: ");
scanf("%d", &n);
printf("\nN = %d\n\n", n);
// 1a. parte
if(verifica_primo(n) == 1)
printf("%d é um número primo!", n);
else
printf("%d não é um número primo!", n);
// 2a. parte
printf("\n\nNúmeros primos até %d:\n\n", n);
tot = 0;
tot_num_linha = 0;
for(r=1; r<=n; r++) {
if(verifica_primo(r) == 1) {
printf("%4d \t", r);
tot=tot+1;
if(tot_num_linha == 9) {
tot_num_linha = 0;
printf("\n");
}
}
else
tot_num_linha = tot_num_linha + 1;
```

```
    }  
  
    }  
  
printf("\n\nTotal de número primos: %d", tot);  
// 3a. parte  
printf("\n\nRelação dos %d primeiros números primos:\n", n);  
tot_num_linha = 0;  
i = 0;  
j = 1;  
while(i<n) {  
if(verifica_primo(j) == 1) {  
printf("%4d \t", j);  
if(tot_num_linha == 9) {  
tot_num_linha = 0;  
printf("\n");  
}  
else  
tot_num_linha = tot_num_linha + 1;  
i = i + 1;  
}  
j = j + 1;  
}  
printf("\n\n");  
return 0;  
}  
FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico, Lógica de Programação. Makron  
Books, São Paulo, 2000. ISBN 857194718X - Capítulo 3
```

Membros da Banca:

Prof. Carlos Norberto Vetorazzi Junior (nome e assinatura)

Prof. Ricardo Ferreira Martins (nome e assinatura)

**Profa Éverlin Fighera Costa Marques
Presidente da Banca (nome e assinatura)**

PROCESSO SELETIVO - 03/2021

Área de Conhecimento: Ciência da Computação/ Algoritmo e Estruturas de Dados

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 4: 1.5 ponto

```
#include <stdio.h>
#include <stdlib.h>
#define D 8
int main()
{
    int mdist[D][D]={0 ,63 ,210,190,0 ,190,181,0},
                {63 ,0 ,160,150,95 ,0 ,0 ,0},
                {210,160, 0, 10,0 ,0 ,104,401},
                {190,150, 10, 0,0 ,0 ,96 , 0},
                {0 ,95 , 0 , 0 , 0 , 80, 0 , 149},
                {190, 0, 0, 0, 80, 0 , 0 , 391},
                {181, 0,104, 96, 0 , 0 , 0 , 0 },
                {0 , 0,401, 0,149,391, 0 , 0}};

    int *rota1, *rota2, M, N, somarota1=0, somarota2=0;
    int lin, col, i, j;
    printf("\nMatriz de Distancias\n");
    printf(" 0.Caceres 1.BBugres 2.Cuiaba 3.VGrande 4.Tangara 5.PLacerda 6.Pocone 7.Parecis\n");
    printf(" \t0|\t1|\t2|\t3|\t4|\t5|\t6|\t7|\n");
    printf(" \t-----");
    for (lin=0; lin<D; lin++){
        printf("\n %i|\t", lin);
        for(col=0; col<D; col++){
            printf(" %i\t",mdist[lin][col]);
        }
    }
    printf("\nDigite as cidades pelos indices da matriz de distancias.\n");
    printf("\nDigite o numero de cidades da primeira rota");
    scanf("%i", &M);
    rota1 = (int*) malloc( M * sizeof(int));
    printf("\nDigite %i cidades da rota 1", M);
    for( i=0; i <M; i++){
        scanf("%i", &rota1[i]);
        if (!(rota1[i]>=0 && rota1[i]<=D) ){
            printf("\nEntrou com cidade invalida ");
            free(rota1);
        }
    }
}
```

```

    return 0;
}
}
for( i=0; i <M-1; i++){
    if ( mdist[rota1[i]][rota1[i+1]]==0){
        printf("\nRota invalida");
        free(rota1);
        return 0;
    } else{
        somarota1 = somarota1 + mdist[rota1[i]][rota1[i+1]];
    }
}
printf("\nRota 1 =\n ");
for( i=0; i <M ; i++){
    printf("%i-", rota1[i]);
}
printf("\nDistancia da rota1 = %i", somarota1);
printf("\nDigite as cidades pelos indices da matriz de distancias.\n");
printf("\nDigite o numero de cidades da segunda rota");
scanf("%i", &N);
rota2 = (int*) malloc( N * sizeof(int));
printf("\nDigite %i cidades da rota 2", N);
for( i=0; i <N; i++){
    scanf("%i", &rota2[i]);
    if (!(rota2[i]>=0 && rota2[i]<=D) ){
        printf("\nEntrou com cidade invalida ");
        free(rota2);
        return 0;
    }
}
for( i=0; i <N-1; i++){
    if ( mdist[rota2[i]][rota2[i+1]]==0){
        printf("\nRota invalida");
        free(rota2);
        return 0;
    } else
        somarota2 = somarota2 + mdist[rota2[i]][rota2[i+1]];
}
printf("\nRota 2 =\n ");
for( i=0; i <N ; i++){
    printf("%i-", rota2[i]);
}
printf("\nDistancia da rota2 = %i", somarota2);

```

```

if ( somarota1 == somarota2){
    printf("\nA distancia eh igual.\n");
}else{
    if ( somarota1 < somarota2){
        printf("\nMenor Rota =\n ");
        for( i=0; i <M ; i++)
            printf("%i-", rota1[i]);

    }else{
        printf("\nMenor Rota =\n ");
        for( i=0; i <N ; i++){
            printf("%i-", rota2[i]);
        }
    }
}
free(rota1);
free(rota2);
return 0;
}

```

FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico, Lógica de Programação. Makron Books, São Paulo, 2000. ISBN 857194718X - Capítulo 3
 SOUZA, Marco Antonio Furlan de et al. Algoritmos e lógica de programação. São Paulo: Thomson, 2005 - Capítulo 5

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

Membros da Banca:

Prof. Carlos Norberto Vetorazzi Junior (nome e assinatura)

Prof. Ricardo Ferreira Martins (nome e assinatura)

**Éverlin Fighera Costa Marques
 Presidente da Banca (nome e assinatura)**

PROCESSO SELETIVO - 03/2021

Área de Conhecimento: Ciência da Computação/Algoritmo e Estruturas de Dados

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 5: 02 pontos (dois)

```
#include<stdio.h>
#define N 5
#define M 5
int main(){
    int matriz[N][M];
    int i, j;
    printf("\nEntre com os valores da matriz: ");
    for( i = 0 ; i < N ; i++ ) {
        printf("\n Linha %d\n", i);
        for( j = 0 ; j < M ; j++ ) {
            printf("(%d)\t", j );
            scanf("%d", &matriz[i][j]);
        }
    }
    // Encontra o maior valor e a linha onde se encontra (i_maior). (40%)
    int i_maior = 0, maior = matriz[0][0];
    for( i = 0 ; i < N ; i++ )
        for( j = 0 ; j < M ; j++ )
            if( matriz[i][j] > maior ){
                i_maior = i;
                maior = matriz[i][j];
            }

    // Procura o menor na linha do maior --> minimax. (40%)
    int j_menor = 0, minimax = matriz[i_maior][0];
    for( j = 0 ; j < M ; j++ )
        if( matriz[i_maior][j] < minimax ){
            j_menor = j;
            minimax = matriz[i_maior][j];
        }
    // Saída dos dados. (incluído nos 20% da entrada)
    printf("Minimax: %d, nas coordenadas (%d, %d)\n", minimax, i_maior, j_menor);
    return 0;
}
```

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de. Algoritmos: lógica para desenvolvimento de programação de computadores. 27. ed. rev. São Paulo: Érica, 2014. Capitulo 7 e 8

SOUZA, Marco Antonio Furlan de et al. Algoritmos e lógica de programação. São Paulo: Thomson, 2005 - Capítulo 5

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

Membros da Banca:

Prof. Carlos Norberto Vetorazzi Junior (nome e assinatura)

Prof. Ricardo Ferreira Martins (nome e assinatura)

Éverlin Fighera Costa Marques
Presidente da Banca (nome e assinatura)

PROCESSO SELETIVO - 03/2021

Área de Conhecimento: Ciência da Computação/Algoritmo e Estruturas de Dados

PROVA ESCRITA - PADRÃO DE RESPOSTA

QUESTÃO 6: 2.5 pontos

```
struct listNode{
    int value;
    struct listNode *next;
};

struct listNode *insert( struct listNode *first, int info){
    struct listNode *newNode = malloc( sizeof(struct listNode) );
    newNode->value = info;
    newNode->next = NULL;
    // Se 'first' for NULL, a lista está vazia --> portanto, retorna 'newNode' (único nó da lista)
    if( first = NULL ){
        return newNode;
    }
    else{ // Percorre até o último nó da lista...
        struct listNode *aux = first;
        while( aux->next != NULL )
            aux = aux->next;
        // Insere 'newNode' no final da lista.
        aux->next = newNode;
        // Retorna o próprio 'first', pois continua sendo o primeiro nó da lista.

        return first;
    }
}
```

TENENBAUM, A.M.; LANGSAM, Y.; AUGENSTEIN, M.J. Estruturas de Dados Usando C. São Paulo: Makron Books, 1995. ISBN 85-346-0348-0

ZIVIANI, Nivio. Projeto de algoritmos: com implementações em Pascal e C. 3. ed. rev. e ampl. São Paulo: Pioneira Thomson Learning, 2011

Membros da Banca:

Prof. Carlos Norberto Vetorazzi Junior (nome e assinatura)

Prof. Ricardo Ferreira Martins (nome e assinatura)

Éverlin Fighera Costa Marques
Presidente da Banca



Assinaturas do documento



Código para verificação: **0TM911TW**

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

- ✓ **EVERLIN FIGHERA COSTA MARQUES** (CPF: 023.XXX.079-XX) em 04/10/2021 às 10:48:28
Emitido por: "SGP-e", emitido em 22/04/2021 - 16:55:11 e válido até 22/04/2121 - 16:55:11.
(Assinatura do sistema)

- ✓ **RICARDO FERREIRA MARTINS** (CPF: 628.XXX.906-XX) em 04/10/2021 às 10:55:49
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:43:00 e válido até 30/03/2118 - 12:43:00.
(Assinatura do sistema)

- ✓ **CARLOS NORBERTO VETORAZZI JUNIOR** (CPF: 070.XXX.978-XX) em 04/10/2021 às 11:21:47
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:34:55 e válido até 30/03/2118 - 12:34:55.
(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link <https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTlwMjJfMDAwMzk0NTlfMzk1NTFFmJyMV8wVE05MTFUVw==> ou o site <https://portal.sgpe.sea.sc.gov.br/portal-externo> e informe o processo **UDESC 00039459/2021** e o código **0TM911TW** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.