UEA - Escola Superior de Tecnologia

Terceira Lista de Exercícios

Disciplina: Programação de Computadores e Algoritmos

Professor: Ricardo Rios

Aluno: Erik Atilio Silva Rey **Matrícula:** 1715310059

Questões

1. Escreva uma função que retorne o k-ésimo dígito (da direita para esquerda) de um inteiro n, k e n dados. Por exemplo, K_esimoDigito(2845, 3) = 8.

R=

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

int main(){
   int n,k;

   int K_esimo(int n, int k);

   printf("Informe N e K:\n");
   scanf("%d%d",&n,&k);

   printf("%d\n",K_esimo(n,k));



   return 0;
}

int K_esimo(int n,int k){
   int a,r,q;
```

```
   for(a=1;a<k;a++){//vai achar o quociente de N em K vezes - 1.
   n = n/10;
   }
   if(r==0){
      printf("Erro\n");
      return -1;
   }
   else{
      r = n%10;//vai achar o resto de N por 10.
      return r;
   }
}
```

2. O fatorial ímpar de um número n ímpar positivo é o produto de todos os números ímpares positivos menores do que ou iguais a n.Indicando o fatorial ímpar de n por n| temos, n|=1.3.5.....n. Por exemplo, 7| = 1 . 3 . 5 . 7 = 105. Escreva funções iterativas e recursivas para a determinação do fatorial ímpar de um inteiro ímpar dado.

Interativa

R=

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

int main(){
   int n;

   int fat(int n);

   printf("Informe um numero impa para o fatorial: \n");
   scanf("%d",&n);
```

```
    fat(n);

    return 0;
}
int fat(int n){
    int r=n,i,soma = 1;

    for(i=1;i<=n;i++){
    if(i%2!=0){
    printf("%d ",i);
    soma*= i;
    }
    }
    printf("= %d",soma);


    return;
}
```

Recurssiva

R=

```
#include <stdio.h>
#include <stdlib.h>
int soma=1;

int main(){
    int n;
```

```
    int fat(int n);

    printf("Informe um numero impa para o fatorial: \n");
    scanf("%d",&n);

    printf("= %d",fat(n));

    return 0;
}
int fat(int n){
int r;

if(n==1){
    printf("%d ",n);
    return 1;
}

else{
    printf("%d ",n);
    r = n*fat(n-2);
}

return r;
}
```

3. Como na questão anterior, o fatorial primo (ou primorial) de um número primo positivo é o produto de todos os primos positivos menores do que ou iguais a ele: p# = 2 . 3 . 5 . 7 . ... .p (sendo 2# = 2). Por exemplo, 7# = 2 . 3 . 5 . 7 = 210. Escreva um programa que determine o fatorial primo de um primo dado.

Interativa

R=

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

int soma=1;
int primorial(int n);

int main(){
   int n;

   printf("Informe n primo para o primorial: \n");
   scanf("%d",&n);

   printf("= %d",primorial(n));



   return 0;
}

int primorial(int n){
   int i,x=n,div = 0,r;

   for (i = 1; i <= x; i++) {
      if (x % i == 0) {
       div++;
      }
      }
```

```
    if (n==1){
        return 1;
    }
    else {

    if (div == 2){
        printf("%d ",n);
        soma *= n;
    }

    primorial(n-1);

    return soma;

    }
}
```

Recurssiva

R=

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

int soma=1;
int primorial(int n);

int main(){
    int n;
```

```
    printf("Informe n primo para o primorial: \n");
    scanf("%d",&n);

    printf("= %d",primorial(n));



    return 0;
}

int primorial(int n){
    int i,x=n,div = 0,r;

    for (i = 1; i <= x; i++) {
        if (x % i == 0) {
         div++;
        }
        }


    if (n==1){
        return 1;
    }
    else {

    if (div == 2){
        printf("%d ",n);
        soma *= n;
    }
```

```
    primorial(n-1);

    return soma;

    }
}
```

4. Escreva funções, iterativa e recursiva, que retornem a soma dos algarismos de um inteiro positivo dado.

Interativa

R=

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

int soma(int n);

int main(){
    int n;

    printf("Informe um inteiro positivo: \n");
    scanf("%d",&n);



    printf(" = %d",soma(n));



    return 0;
}
```

```
int soma(int n){
   int i,a=0;

   for(i=1;i<=n;i++){
   printf("%d",i);
   a += i;
   if(i!=5){
   printf(" + ");
       }
   }

   return a;
}
```

Recurssiva

R=

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

int soma(int n);

int main(){
   int n;

   printf("Informe um inteiro positivo: \n");
   scanf("%d",&n);


   printf(" = %d",soma(n));
```

```
    return 0;
}
int soma(int n){
    int i,a;

    if (n==1){
    printf("1");
    return 1;
    }
    else{
    printf("%d + ",n);
    a = n+soma(n-1);
    }

    return a;
}
```

5. Escreva uma função recursiva que retorne o n-ésimo termo da sequência de Fibbonaci, n dado.

R=

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

int fibo(int n);

int main(){
```

```
    int n;

    printf("Informe o termo que fibonacci: \n");
    scanf("%i",&n);

    fibo(n);

    return 0;
}
int fibo(int n){
    int i,a=1,b=1,c;

    if (n==1){
    printf("%d ",a);
    }
    else if(n==2){
    printf("%d ",a);
    printf("%d ",b);
    }
    else{
    printf("%d ",a);
    printf("%d ",b);
    for(i=1;i<n-1;i++){
    c=a+b;
    printf("%d ",c);
    a=b;b=c;
        }
    }
}
```

6. Escreva uma função recursiva que gere uma tabuada de multiplicação, exibindo-a no formato (para posicionar a saída pode-se utilizar a função gotoxy()).

1x2 = 2 1x3 = 3 1x4 = 4 ...

2x2 = 4 2x3 = 6 2x4 = 8 ...

3x2 = 6 3x3 = 9 3x4 = 12 ...

... ...

9x2 = 6 9x3 = 27 9x4 = 36 …

7. Escreva uma função recursiva que determine o mínimo múltiplo comum de dois inteiros dados.

R=

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

int mdc(int a,int b);

int main(){
   int a,b;

   printf("informe A e B: \n");
   scanf("%d%d",&a,&b);

   printf("O mdc de %d e %d é = %d\n",a,b,mdc(a,b));

   return 0;
}
int mdc(int a,int b){
   int r,resultado;
```

```
    if(a%b==0){
    return b;
    }
    else{
    mdc(a,a%b);
    }

}
```

8. Escreva funções, recursiva e iterativa, que implementem a função pow().

Interativa

R=

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

int pot(int a,int e);

int main(){
    int a,e;

    printf("Digite a base e o expoente:\n");
    scanf("%i%i",&a,&e);

    printf("%i elevado a %i é = %i\n",a,e,pot(a,e));

    return 0;
}
```

```
int pot(int a,int e){
   int r=1,i;

   for(i=0;i<e;i++){
   r*=a;
   }
   return r;
}
```

Recursiva

R=

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

int pot(int a,int e);

int main(){
   int a,e;

   printf("Digite a base e o expoente:\n");
   scanf("%i%i",&a,&e);

   printf("%i elevado a %i é = %i\n",a,e,pot(a,e));

   return 0;
}

int pot(int a,int e){
   int r;
```

```
    if(e==1){
    return a;
    }
    else{
    r = a*pot(a,e-1);
    }
    return r;
}
```

9. Escreva uma função recursiva que retorne o maior elemento de um vetor.

R=

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

int vet(int a[],int n);

int main(){
    int n;

    printf("Informe N:\n");
    scanf("%d",&n);

    int a[n],i;

    printf("Digite os valores inteiro do vetor:\n");

    for(i=0;i<n;i++){
    printf("");
```

```
    scanf("%i",&a[i]);
    }

    printf("O maior valor é %d.",vet(a,n));

    return 0;
}
int vet(int a[],int n){

    if(n==0){
    return 0;
    }
    else if(a[n-1]>vet(a,n-1)){
    return a[n-1];
    }else{
    return vet(a,n-1);
    }
}
```

10.Escreva uma função que exiba as componentes de um vetor na ordem inversa daquela em que foram armazenadas.

R=

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

int inv(int a[],int n);

int main(){
    int n;

    printf("Informe N:\n");
    scanf("%d",&n);
```

```
   int a[n],i;

   printf("Digite os valores inteiro do vetor:\n");

   for(i=0;i<n;i++){
   printf("");
   scanf("%i",&a[i]);
   }

   printf("A ordem inversa é:\n");

   inv(a,n);


   return 0;
}
int inv(int a[],int n){
   int g=n,i;

   for(i=0;i<n;i++){
   printf("%d ",a[g-1]);
   g--;
   }
}
```

11. Um vetor é palíndromo se ele não se altera quando as posições das componentes são invertidas. Por exemplo, o vetor v = {1, 3, 5, 2, 2, 5, 3, 1} é palíndromo. Escreva uma função que verifique se um vetor é palíndromo.

R=

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


int ver(int a[],int n);


int main(){
```

```
    int n;

    printf("Informe N:\n");
    scanf("%d",&n);

    int a[n]; int i;

    printf("Digite os valores inteiros do vetor:\n");
    for(i=0;i<n;i++){
    scanf("%d",&a[i]);
    }

    ver(a,n);

    return 0;
}
int ver(int a[],int n){
    int b[n],i,g=n-1,p=0,x1,x2;

    for(i=0;i<n;i++){
        b[i]=a[g];
        g--;
    }
        g=n-1;

    for(i=0;i<n;i++){
        x1=b[i];
        x2=a[g];
        if(x1 == x2) p++;
```

```
    g--;
  }
  if(p==n) printf("Eh palindromo.\n");
  else printf("Não eh palindromo.\n");



}
```

12. Escreva uma função que receba um vetor e o decomponha em dois outros vetores, um contendo as componentes de ordem ímpar e o outro contendo as componentes de ordem par. Por exemplo, se o vetor dado for v = {3, 5, 6, 8, 1, 4, 2, 3, 7}, o vetor deve gerar os vetores u = {3, 6, 1, 2, 7} e w = {5, 8, 4, 3}.

R=

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

int ver(int a[],int n);

int main(){
  int n;

  printf("Informe N:\n");
  scanf("%d",&n);

  int a[n];
  register int i;

  printf("Digite os valores inteiros do vetor:\n");
  for(i=0;i<n;i++){
  scanf("%d",&a[i]);
```

```
    }

    ver(a,n);

    return 0;
}
int ver(int a[],int n){
    register int i;
    int par[n];

    for(i=0;i<n;i++){
        if(a[i]%2==0) par[i]=a[i],a[i]=0;
        else par[i]=0;
    }

        printf("Pares:\n");
    for(i=0;i<n;i++){
        if(par[i]!=0) printf("%d ",par[i]);
    }

        printf("\n");

        printf("Impares:\n");
    for(i=0;i<n;i++){
        if(a[i]!=0) printf("%d ",a[i]);
    }


}
```

13. Um vetor do Rn é uma n-upla de números reais v = {x_1, x_2, ..., x_n}, sendo cada xi chamado de componente. A norma de um vetor v = {x_1, x_2, ..., x_n} é definida por

Escreva uma função que receba um vetor do Rn, n dado, e forneça sua norma.

R=

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

int norma(float a[],int n);

int main(){
   int n;

   printf("Informe N:\n");
   scanf("%i",&n);

   float a[n];
   register int i;

   printf("Digite os valores reais do vetor:\n");
   for(i=0;i<n;i++){
   scanf("%f",&a[i]);
   }

   printf("A norma do vetor eh = %2.f\n",norma(a,n));

   return 0;
}
int norma(float a[],int n){
```

```
float b[n],soma=0;
register int i;

//X²
for(i=0;i<n;i++){
b[i] = (a[i])*(a[i]);
}

//X1+X2+...+Xn
for(i=0;i<n;i++){
soma += b[i];
}

//raiz da soma
soma = pow(soma,0.5);

return soma;
}
```

14. O produto escalar de dois vetores do Rn é a soma dos produtos das componentes correspondentes. Isto é, se u = {x_1, x_2, ..., x_n} e v = {y_1, y_2, ..., y_n}, *o produto escalar é* $x_1.y_1 + x_2.y_2 ... + x_n.y_n$. Escreva uma função que receba dois vetores do Rn, n dado, e forneça o produto escalar deles.

R=

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

int norma(float a[],float b[],int n);
```

```
int main(){
    int n;

    printf("Informe N:\n");
    scanf("%i",&n);

    float a[n],b[n];
    register int i;

    printf("Digite os valores reais do vetor A:\n");
    for(i=0;i<n;i++){
    scanf("%f",&a[i]);
    }
    printf("Digite os valores reais do vetor B:\n");
    for(i=0;i<n;i++){
    scanf("%f",&b[i]);
    }

    printf("O produto escalar dos vetores eh = %.2f\n",escalar(a,b,n));

    return 0;
}
int escalar(float a[],float b[],int n){
     register int i;
     float c[n],soma=0;

     for(i=0;i<n;i++){
        c[i]=(a[i])*(b[i]);
     }
```

```
    for(i=0;i<n;i++){
    soma += c[i];
    }

    return soma;
}
```

15. A amplitude de uma relação de números reais é a diferença entre o maior e o menor valores da relação. Por exemplo, a amplitude da relação 5, 7, 15, 2, 23 21, 3, 6 é 23 - 2 = 21. Escreva uma função que receba uma relação de números e forneça sua amplitude.

R=

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

int amplitude(float a[],int n);

int main(){
   int n;

   printf("Informe N:\n");
   scanf("%i",&n);

   float a[n];
   register int i;

   printf("Digite os valores reais do vetor A:\n");
   for(i=0;i<n;i++){
   scanf("%f",&a[i]);
   }
```

```
    printf("A amplitude do vetor eh = %d\n",amplitude(a,n));

    return 0;
}
int amplitude(float a[],int n){
     register int i,maior=a[1],menor=a[0];
     double amp=0;

     for(i=0;i<n;i++){
     if(a[i] < menor) menor=a[i];
     if(a[i] > maior) maior=a[i];
      }

     amp = maior - menor;

     return amp;
}
```

16. O desvio padrão de uma relação de números reais é a raiz quadrada da média aritmética dos quadrados dos desvios. Escreva uma função que receba uma relação de números reais e forneça o seu desvio padrão.

R=

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

int desvio(float vet[],int n);

int main(){
    int n;
```

```
    printf("Informe n:\n");
    scanf("%i",&n);

    float vet[n];

    printf("Informe os valores reais: \n");
    for(int i=0;i<n;i++){
        scanf("%f",&vet[i]);
    }

    desvio(vet,n);



    return 0;
}
int desvio(float vet[],int n){
    float resultado=0,soma=0,soma2=0,a;

    for(int i=0;i<n;i++){
    soma += vet[i];
    a = vet[i]*vet[i];
    soma2 += a;
    }

    printf("Soma = %.2f\nSoma2 = %.2f\n",soma,soma2);
    resultado = (soma2 - (pow(soma,2)/n));
    resultado = (resultado/(n-1));
    resultado = sqrt(resultado);
```

```
   printf("O desvio padrão é = %.2f \n",resultado);
}
```

17. Escreva uma função que forneça as componentes distintas de um vetor dado. Por exemplo, se o vetor dado for v = {3, 2, 1, 3, 4, 1, 5, 5, 2} a função deve fornecer v = {3, 2, 1, 4, 5}.

R=

```
#include <stdio.h>
#include <stdlib.h>
#define tam 9int v[10];
int v_dis[10];
int i, j, repete = 0, n = 0;
int main(){
int v[10];
        for (i = 0; i < 10; i++){
        scanf("%d",&v[i]);
      if (i == 0){
        v_dis[n] = v[i];
        n++;
        } else {
        repete = 0;
        for (j = 0; j < n; j++){
        if(v[i] == v_dis[j]){
        repete++;
        }}
        if (repete < 1) {
        v_dis[n] = v[i];
        n++;
        }}}
```

```
        printf("Vetor Distinto: ");
        for (i = 0; i < n; i++) printf("%d ", v_dis[i]);
    return 0;
}
```

18. Algumas empresas que realizam sorteios de prêmios entre seus clientes o fazem através dos sorteios da loteria federal, sendo ganhador o número formado pelos algarismos das casas das unidades dos números sorteados nos cinco prêmios da referida loteria. Por exemplo, se o sorteio da loteria federal deu como resultado os números 23451, 00234, 11236, 01235 e 23452, o prêmio da tal empresa seria dado ao cliente que possuísse o bilhete de número 14652. Escreva uma função que receba os números sorteados pela loteria federal e forneça o número que ganhará o prêmio de acordo com as regras acima.

R=

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

void sorte(int vet[5][5]);

int main(){
    int vet[5][5];

    printf("Digite os números sorteados");
    for(int i=0;i<5;i++){
    printf("\nDigite o %d numero:\n",i+1);
    for(int j=0;j<5;j++){
        scanf("%d",&vet[i][j]);
    }
    }

    sorte(vet);
```

```
return 0;
}
void sorte(int vet[5][5]){
int vet_g[5];

    vet_g[0] = vet[0][4];
    vet_g[1] = vet[1][4];
    vet_g[2] = vet[2][4];
    vet_g[3] = vet[3][4];
    vet_g[4] = vet[4][4];

    printf("\nO número ganhador: \n");
    for(int i=0;i<5;i++){
    printf("%d ",vet_g[i]);
    }

}
```

19. Escreva uma função que insira um valor dado num vetor numa posição dada. Por exemplo, se o vetor for v = {3, 8, 5, 9, 12, 3}, o valor dado for 10 e a posição dada for 4, a função deve fornecer v = {3, 8, 5, 10, 9, 12, 3}.

R=

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

void muda(int vet[],int x,int m);

int main(){
```

```
int n,x,i,m;

printf("Digite o tamanho do vetor:\n");
scanf("%i",&n);

int vet[n];

printf("Digite os valores do vetor abaixo:\n");
for(i=0;i<n;i++){
scanf("%i",&vet[i]);
}

printf("O vetor é:\nVetor = {");
for(i=0;i<n;i++){
   if(i==n-1){
      printf(" %d ",vet[i]);
      }else
printf(" %d,",vet[i]);

}
printf("}\n");

printf("Digite a posição a ser mudada e o valor: ");
scanf("%d %d",&x,&m);
muda(vet,x,m);

printf("\n");

printf("Novo vetor:\nVetor = {");
```

```
    for(i=0;i<n;i++){
        if(i==n-1){
            printf(" %d ",vet[i]);
            }else
    printf(" %d,",vet[i]);
    }
    printf("}\n");



return 0;
}
void muda(int vet[],int x,int m){

    vet[x-1] = m;

    return vet;
}
```

20. Escreva uma função que insira um valor dado num vetor ordenado de modo que o vetor continue ordenado. Por exemplo, se o vetor dado for v = {2, 5, 7, 10, 12, 13} e o valor dado for 6, a função deve fornecer o vetor v = {2, 5, 6, 7, 10, 12, 13}.

R=

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

void muda(int vet[],int x,int m);
void ordem(int vet[],int n);

int main(){
    int n,x,i,m;
```

```
printf("Digite o tamanho do vetor:\n");
scanf("%i",&n);

int vet[n];

printf("Digite os valores do vetor abaixo:\n");
for(i=0;i<n;i++){
scanf("%i",&vet[i]);
}

printf("O vetor é:\nVetor = {");
for(i=0;i<n;i++){
   if(i==n-1){
      printf(" %d ",vet[i]);
      }else
printf(" %d,",vet[i]);

}
printf("}\n");

printf("Digite a posição a ser mudada e o valor: ");
scanf("%d %d",&x,&m);
muda(vet,x,m);
ordem(vet,n);

printf("\n");

printf("Novo vetor em ordem crescente:\nVetor = {");
```

```
    for(i=0;i<n;i++){
        if(i==n-1){
            printf(" %d ",vet[i]);
            }else
    printf(" %d,",vet[i]);
    }
    printf("}\n");


return 0;
}
void muda(int vet[],int x,int m){

    vet[x-1] = m;



    return vet;
}
void ordem(int vet[],int n){
    int i,f,aux;

    for(i=0;i<n;i++){
        for(f=i+1;f<n;f++){
            if(vet[i]>vet[f]){
                aux = vet[i];
                vet[i] = vet[f];
                vet[f] = aux;
            }
        }
```

```
    }

    return vet;
}
```

21.Escreva uma função que remova uma componente de ordem dada de um vetor dado. Por exemplo, se o vetor dado for v = {2, 5, 7, 10, 12, 13} e a componente a ser removida for a de ordem 4, programa deve fornecer o vetor v = {2, 5, 7, 12, 13}.

R=

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

void remo(int vet[],int x,int n);

int main(){
    int n,x,i,m;

    printf("Digite o tamanho do vetor:\n");
    scanf("%i",&n);

    int vet[n];

    printf("Digite os valores do vetor abaixo:\n");
    for(i=0;i<n;i++){
    scanf("%i",&vet[i]);
    }

    printf("O vetor é:\nVetor = {");
    for(i=0;i<n;i++){
        if(i==n-1){
```

```
            printf(" %d ",vet[i]);
            }else
    printf(" %d,",vet[i]);

    }
    printf("}\n");

    printf("Digite a posição a ser removida: ");
    scanf("%d",&x);
    remo(vet,x,n);

return 0;
}
void remo(int vet[],int x,int n){

    vet[x-1] = 0;
    int i;

printf("Novo vetor:\nVetor = {");
    for(i=0;i<n;i++){
       if(vet[i]!= 0){
           if(i == n-1){
            printf(" %d ",vet[i]);
           }else{
       printf(" %d,",vet[i]);
       }
       }
    }
    printf("}\n");
```

```
}
```

22. Escreva uma função que, dadas duas relações de números, cada uma delas com números distintos, forneça os números que aparecem nas duas listas. Por exemplo, se as relações forem u = {9, 32, 45, 21, 56, 67, 42, 55} e w = {24, 42, 32, 12, 45, 11, 67, 66, 78}, a função deve fornecer o vetor v = {32, 45, 67, 42}.

R=

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

void dist(int vet1[],int vet2[],int n1,int n2);

int main(){
   int n1,i;

   printf("Digite o tamanho do vetor:\n");
   scanf("%i",&n1);

   int vet1[n1];

   printf("Digite os valores do vetor1 abaixo:\n");
   for(i=0;i<n1;i++){
   scanf("%i",&vet1[i]);
   }

   printf("O vetor1 é:\nVetor1 = {");
   for(i=0;i<n1;i++){
```

```
   if(i==n1-1){
       printf(" %d ",vet1[i]);
       }else
printf(" %d,",vet1[i]);

}
printf("}\n");

int n2;
printf("Digite o tamanho do vetor2:\n");
scanf("%i",&n2);

int vet2[n2];

printf("Digite os valores do vetor2 abaixo:\n");
for(i=0;i<n2;i++){
scanf("%i",&vet2[i]);
}

printf("O vetor2 é:\nVetor2 = {");
for(i=0;i<n2;i++){
   if(i==n2-1){
       printf(" %d ",vet2[i]);
       }else
printf(" %d,",vet2[i]);

}
printf("}\n");
```

```
    dist(vet1,vet2,n1,n2);



return 0;
}
void dist(int vet1[],int vet2[],int n1,int n2){
    int vet3[n1+n2],i;

    for(i=0;i<n1+n2;i++){
        if(i<=n1) vet3[i]=vet1[i];
        else vet3[i]=vet2[i];
    }
    int repete=0,j;

    for(i=0;i<n1+n2;i++){
        for(j=i;j<n1+n2;j++){
            if(vet3[i] == vet3[j]) repete+=1;
        }
    }

    int nr=0,vetR[repete];

    for(i=0;i<n1+n2;i++){
        for(j=i;j<n1+n2;j++){
            if(vet3[i] == vet3[j]){
            vetR[nr] = vet3[i];
            nr+=1;
            }
        }
```

```
    }

    printf("O vetor distinto é:\nVetor = {");
    for(i=0;i<nr;i++){
        if(i==nr-1){
            printf(" %d ",vetR[i]);
            }else
    printf(" %d,",vetR[i]);

    }
    printf("}\n");

}
```

23. Uma avaliação escolar consiste de 50 questões objetivas, cada uma delas com 5 opções, v = {1, 2, 3, 4 e 5}, sendo apenas uma delas verdadeira. Escreva uma função que receba a sequência de respostas corretas, o gabarito, e corrija um cartão-resposta dado.

R=

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

void corrige(int gabarito[],int respostas[]);

int main(){
    int tam=50,i,gabarito[50],respostas[50];
    int  aux;

    printf("Digite o gabarito (1-a | 2-b | 3-c | 4-d | 5-e):\n");
    for(i=0;i<50;i++){
        printf("Qual o gabarito da questão %d: ",i+1);
```

```
        scanf("%d",&aux);
        if(aux > 5 || aux < 0){
        printf("Você digitou errado!\nDigite novamente a questão %d:",i+1);
            while(aux > 5 || aux < 0){
                printf("Qual o gabarito da questão %d: ",i+1);
                scanf("%d",&aux);
            }
            gabarito[i]=aux;
        }else{
        gabarito[i]=aux;
        }
    }

    printf("Digite suas respostas (1-a | 2-b | 3-c | 4-d | 5-e):\n");
    for(i=0;i<50;i++){
        printf("Questão %d R = ",i+1);
        scanf("%d",&respostas[i]);
    }

    corrige(gabarito,respostas);

return 0;
}
void corrige(int gabarito[],int respostas[]){
    int total=0,i;

    for(i=0;i<50;i++){
        if(gabarito[i]==respostas[i]){
        total+=1;
```

```
        }
    }
    printf("Total de acertos = %d\n",total);
}
```

24. Escreva uma função que forneça o valor numérico de um polinômio P(x) dado, para um valor de x dado. Por exemplo, se o polinômio dado for P(x) = x3 + 2x - 1 e o valor de x dado for 2, a função deve fornecer P(2) = 23 + 2x2 - 1 = 11.

R=

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

int poli(char *vet,char x);

int main(){
    char *vet;
    vet = malloc(10);

    printf("informe o polinomio:\n");
    gets(vet);

    char x;
    printf("Informe o X:\n");
    scanf("%s",&x);

    for(int t=0;t<=strlen(vet)-1;t++){
    if(vet[t]=='x') vet[t]=x , printf("*") , putchar(vet[t]);
    else putchar(vet[t]);
    }
```

```
    free(vet);
    printf("\n");


return 0;
}
```

25. Escreva uma função que forneça a transposta de uma matriz dada.

R=

ex: 3x2

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

void at(int i,int j);

int main(){
    int i,j;

    printf("Digite a quantidade de linhas e a colunas da matriz: ");
    scanf("%i%i",&i,&j);

    at(i,j);

return 0;
}
void at(int i,int j){
int matriz[i][j];
    printf("Preencha a matriz:\n");
```

```
    for(int a=0;a<i;a++){
        for(int b=0;b<j;b++){
            printf("Digite a %d linha e %d coluna: ",a+1,b+1);
            scanf("%i",&matriz[a][b]);
        }
    }
    printf("\nA matriz digitada foi:\n");
    for(int a=0;a<i;a++){
        for(int b=0;b<j;b++){
        printf("%d ",matriz[a][b]);
        }
        printf("\n");
    }
    printf("\nA transposta é:\n");
    for(int a=0;a<j;a++){
        for(int b=0;b<i;b++){
        printf("%d ",matriz[b][a]);
        }
        printf("\n");
    }
}
```

26.Uma matriz quadrada é dita triangular se os elementos situados acima de sua diagonal principal são todos nulos. Escreva uma função que receba uma matriz quadrada e verifique se ela é triangular.

R=

ex: 3x3

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

```
int main(){
    int i,j;

    printf("Digite a quantidade de linhas e a colunas da matriz: ");
    scanf("%i%i",&i,&j);

    int matriz[i][j];
    printf("Preencha a matriz:\n");
    for(int a=0;a<i;a++){
        for(int b=0;b<j;b++){
            printf("Digite a %d linha e %d coluna: ",a+1,b+1);
            scanf("%i",&matriz[a][b]);
        }
    }
    printf("\nA matriz digitada foi:\n");
    for(int a=0;a<i;a++){
        for(int b=0;b<j;b++){
        printf("%d ",matriz[a][b]);
        }
        printf("\n");
        }

    int vdd=0,fat=0;
    for(int a=0;a<i;a++){
        for(int b=0;b<j;b++){
        if(b>a){
            if(matriz[a][b]== 0) vdd+=1;
        }
    }
```

```
    }
    for(int t=0;t<j;t++){
    fat+=t;
    }
    if(vdd==fat) printf("é triangular!\n");
    else printf("Não é triangular!\n");

    return 0;

}
```

27. Escreva uma função que determine o produto de duas matrizes.

R=

ex:10x4 . 4x3

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

int main(){
    int i, j, lA, cA, lB, cB, x;
    lA=10;
    cA=4;
    lB=4;
    cB=3;
    float matrizA[lA][cA],matrizB[lB][cB],matrizC[lA][cB],Aux=0;
        for(i=0; i<10; i++){
            for(j=0; j<4; j++){
                printf("Informe os Componentes da Matriz A(por linha): ");
                scanf("%f", &matrizA[i][j]);
```

```
        }
        printf("\n");
    }
    for(i=0; i<4; i++){
        for(j=0; j<3; j++){
            printf("Informe os componentes da Matriz B(por linha): ");
            scanf("%f", &matrizB[i][j]);
        }
        printf("\n");
    }
printf("Matriz A \n\n");
    for(i=0; i<10; i++){
        for(j=0; j<4; j++){
            printf("%6.f", matrizA[i][j]);
        }
        printf("\n\n");
    }
printf("Matriz B \n\n");
    for(i=0; i<4; i++){
        for(j=0; j<3; j++){
            printf("%6.f", matrizB[i][j]);
        }
        printf("\n\n");
    }
    for(i=0; i<10; i++){
        for(j=0; j<3; j++){
            matrizC[i][j]=0;
                for(x=0; x<4; x++){
                    Aux+= matrizA[i][x] * matrizB[x][j];
```

```
                }
                matrizC[i][j]=Aux;
                Aux=0;
            }
        }
    printf("\n\n");
    printf("Matriz Gerada da Multiplicacao A*B \n\n");
        for(i=0; i<10; i++){
            for(j=0; j<3; j++){
                printf("%6.f", matrizC[i][j]);
            }
            printf("\n\n");
        }
    printf("\n\n");
    return 0;
}
```