# Algoritmos Computacionais

Sumário

[Algoritmos Computacionais 1](#_Toc67833987)

[Estruturas Condicionais - if 2](#_Toc67833988)

[Estruturas Condicionais – Operador Ternário 19](#_Toc67833989)

[Exercícios Condicionais 2 21](#_Toc67833990)

[Estruturas de Repetição – For 33](#_Toc67833991)

[Estruturas de Repetição – While 43](#_Toc67833992)

[Estruturas de Repetição – do while 62](#_Toc67833993)

[Estrutura de Dados Homogêneos – Vetor 72](#_Toc67833994)

[Estruturas de dados homogêneos - Matriz 79](#_Toc67833995)

[Modularização - Função 88](#_Toc67833996)

[Passagem de parâmetros 98](#_Toc67833997)

[Recursão 105](#_Toc67833998)

[Estrutura de Dados Heterogêneas - Registro 120](#_Toc67833999)

[Estruturas de dados dinâmicas - Ponteiro 132](#_Toc67834000)

[Listas simplesmente encadeadas 142](#_Toc67834001)

## Estruturas Condicionais - if

1. Faça um programa que leia um número e diga se ele é par ou ímpar.

int x;

main(){

printf("Diga um numero:\n");

scanf("%d",&x);

if(x%2==0)

printf("%d e par.",x);

else

printf("%d e impar.",x);

}

1. Faça um programa que leia um número, e se ele for maior que 20 então imprima metade do número, caso contrário imprima o dobro do valor.

float n;

main(){

printf("Diga o numero:\n");

scanf("%f",&n);

if (n>20){

printf("O numero %.2f e maior que vinte, entao virou %.2f",n,n/2);

}

else{

printf("O numero %.2f e maior que 20, entao virou %.2f",n,n\*2);

}

}

1. Faça um programa que leia um número e diga se esse número é positivo, negativo ou nulo.

float n;

main(){

printf("Diga o numero:\n");

scanf("%f",&n);

if(n>0)

printf("%.2f e positivo.",n);

else{

if(n<0)

printf("%.2f e negativo.",n);

else

printf("%.2f e nulo.",n);

}

}

1. Faça um programa que imprima a raiz quadrada de um número caso ele seja positivo e o quadrado do número caso ele seja negativo.

float x;

main(){

printf("Diga o numero\n");

scanf("%f",&x);

if(x>0)

printf("A raiz de %.2f e %.2f",x,pow(x,0.5));

if(x<0)

printf("A potencia de %.2f e %.2f",x,pow(x,2));

if(x==0)

printf(“O numero tem valor nulo.”);

}

1. Faça um programa que leia um peso no planeta Terra e o número de um planeta e imprima o valor do seu peso neste planeta. A relação dos planetas é dada a seguir juntamente com o valor das gravidades relativas ao planeta Terra.

float peso;

int planeta;

main(){

printf("Diga seu peso e o numero do planeta:\n");

scanf("%f%d",&peso,&planeta);

if(planeta==1)

printf("seu peso em mercurio e %.2f",peso\*0.37);

if(planeta==2)

printf("seu peso em venus e %.2f",peso\*0.88);

if(planeta==3)

printf("seu peso em marte e %.2f",peso\*0.38);

if(planeta==4)

printf("seu peso em jupiter e %.2f",peso\*2.64);

if(planeta==5)

printf("seu peso em saturno e %.2f",peso\*1.15);

if(planeta==6)

printf("seu peso em urano e %.2f",peso\*1.17);

if(planeta<=0||planeta>6)

printf(“Esse nao e um valor valido.”);

}

1. Faça um programa que leia um caractere alfanumérico e diga se ele é uma vogal.

char x;

main(){

printf("diga se uma letra.\n");

scanf("%c",&x);

x=tolower(x);

if(x=='a'||x=='e'||x=='i'||x=='o'||x=='u')

printf("%c e vogal",x);

else

printf("%c e uma consoante",x);

}

1. Faça um programa que leia um caractere alfanumérico e diga se ele é consoante, número ou vogal.

char a;

main(){

printf("Diga o caracter alfa numerico:\n");

scanf("%c",&a);

a=tolower(a);

if(a=='a'||a=='e'||a=='i'||a=='o'||a=='u')

printf("%c e vogal",a);

else

if(a>='b'&&a<='z')

printf("%c e uma consoante",a);

else

if(a>='0'&&a<='9')

printf("%c e um numero",a);

else

printf("outros casos");

}

1. Faça um programa que peça para o usuário entrar com uma senha e diga se a senha está correta ou incorreta. A senha é definida como uma constante pelo programador.

#define SENHA "senha"

char s[13];

main(){

printf("insira a senha.\n");

scanf("%s",&s);

if(strcmp(s,senha))

printf("senha incorreta");

else

printf("senha correta");

}

1. Pesquise as funções strcpy(), strcat(), strlen(). Diga para que servem e como são usadas. Dê exemplos.

//Primeiro, strcpy(). Essa funcao serve para copiar o

//conteudo de uma string para a outra. Por exemplo,

//strcpy(colar,copiar) vai copiar o conteudo da string

//"copiar" e colar na string "colar". Ex no código:

char colar [11],copiar[11];

main(){

printf("Diga a palavra que vai ser copiada e depois a substituida\n");

scanf("%s%s",&copiar,&colar);

strcpy(colar,copiar);

printf("%s %s, como ve, a palavra foi substituida.",colar, copiar);

}

//Agora, strcat(). Essa funcao junta duas strings

//em apenas uma, conforme o exemplo a seguir:

//strcat(base,juntar), essa funcao vai adicionar

//a string "base" o conteudo da string "juntar".

char base [51],juntar[51];

main(){

printf("Primeiro, diga uma palvra.\n");

scanf("%s",&base);

printf("%s? Otimo, agora diga outra.\n",base);

scanf("%s",&juntar);

strcat(base,juntar);

printf("%s. Como ve, as duas se juntaram em uma.",base);

}

//Por fim, strlen(). Essa funcao diz a quantidade

//de letras de uma string, que fica armazenada em

//uma outra variavel. Conforme o exemplo abaixo.

char palavra[21];

int tamanho;

main(){

printf("Diga uma palavra.\n");

scanf("%s",&palavra);

tamanho=strlen(palavra);

printf("Essa palavra tem %d letras",tamanho);

}

1. Faça um programa que leia um número e diga se ele é ou não é múltiplo de 3.

int m;

main(){

printf("Diga um numero:\n");

scanf("%d",&m);

if(m%3)

printf("%d nao e multiplo de 3",m);

else

printf("%d e multiplo de 3",m);

}

1. Faça um programa que leia um número e diga se ele é ou não é divisível por 5.

int m;

main(){

printf("Diga um numero:\n");

scanf("%d",&m);

if(m%5)

printf("%d nao e multiplo de 5",m);

else

printf("%d e multiplo de 5",m);

}

1. Faça um programa que leia um número e diga se ele é divisível por 3 e por 7. Obs.: utilize operador lógico.

int x;

main(){

printf("diga o numero:\n");

scanf("%d",&x);

if((x%3==0)&&(x%7==0))

printf("%d e multiplo de 3 e de 7",x);

else

printf("%d nao e multiplo de 3 e 7.",x);

}

1. Faça um programa que leia um número e diga se ele é divisível por 3 e por 7. Obs.: Não é permitido a utilização de operador lógico. Se o número for divisível por 21 ele então é divisível por 3 e por 7.

int x;

main(){

printf("diga o numero:\n");

scanf("%d",&x);

if(x%21)

printf("%d nao e multiplo de 3 e 7.",x);

else

printf("%d e multiplo de 3 e de 7",x);

}

1. Faça um programa que leia um número e diga se ele está compreendido entre 20 e 90 ou não.

float x;

main(){

printf("diga o numero:\n");

scanf("%f",&x);

if((x>=20)&&(x<=90))

printf("%.2f esta entre 20 e 90",x);

else

printf("%.2f nao esta entre 20 e 90",x);

}

1. Faça um programa que leia a sigla do estado em que uma pessoa nasceu e imprima uma das mensagens abaixo: •carioca •paulista •mineiro •baiano •outros casos

char sigla[3];

main(){

printf("diga a sigla do seu estado:\n");

scanf("%s",&sigla);

if(!(strcmp(sigla,"RJ")))

printf("Carioca");

else

if(!(strcmp(sigla,"SP")))

printf("Paulista");

else

if(!(strcmp(sigla,"MG")))

printf("Mineiro");

else

if(!(strcmp(sigla,"BA")))

printf("Baiano");

else

printf("Outros casos.");

}

1. Faça um programa que leia um número inteiro de 3 dígitos e informe se o algarismo da casa das centenas é par ou ímpar.

int x;

main(){

printf("Diga um numero inteiro de 3 digitos:\n");

scanf("%d",&x);

if(x<1000){

x=x/100;

if(x%2)

printf("%d, o numero da casa das centenas, e impar. ",x);

else

printf("%d, o numero da casa das centenas, e par. ",x);

}

else

printf("%d tem mais de 3 digitos.",x);

}

1. Faça um programa que leia 2 palavras e diga se elas são iguais ou diferentes.

char palavra [10],word[10];

main(){

printf("diga duas palavras\n");

scanf("%s%s",&palavra,&word);

if(strcmp(palavra,word))

printf("sao duas palavras diferentes.");

else

printf("sao duas palavras iguais.");

}

1. Faça um programa que leia 3 números diferentes e os imprima em ordem crescente e decrescente. Se houver números iguais exibir mensagem de erro.

float x,y,z;

main(){

printf("Diga os tres numeros diferentes.\n");

scanf("%f%f%f",&x,&y,&z);

if((x==y)||(x==z)||(y==z))

printf("Os numeros devem ser diferentes.");

else

if((x>y)&&(x>z)&&(y>z))

printf("%.2f, %.2f, %.2f, %.2f, %.2f, %.2f",x,y,z,z,y,x);

else

if((x>y)&&(x>z)&&(z>y))

printf("%.2f, %.2f, %.2f, %.2f, %.2f, %.2f",x,z,y,y,z,x);

else

if((y>x)&&(y>z)&&(x>z))

printf("%.2f, %.2f, %.2f, %.2f, %.2f, %.2f",y,x,z,z,x,y);

else

if((y>x)&&(y>z)&&(z>x))

printf("%.2f, %.2f, %.2f, %.2f, %.2f, %.2f",y,z,x,x,z,y);

else

if((z>x)&&(z>y)&&(x>y))

printf("%.2f, %.2f, %.2f, %.2f, %.2f, %.2f",z,x,y,y,x,z);

else

if((z>x)&&(z>y)&&(y>x))

printf("%.2f, %.2f, %.2f, %.2f, %.2f, %.2f",z,y,x,x,y,z);

}

1. Faça um programa que leia 5 números e identifique o maior e o menor.

float a,b,c,d,e,maior,menor;

main(){

printf("Diga 5 numeros.\n");

scanf("%f%f%f%f%f",&a,&b,&c,&d,&e);

maior=a;

menor=a;

if(b>maior)

maior=b;

if(c>maior)

maior=c;

if(d>maior)

maior=c;

if(e>maior)

maior=e;

if(b<menor)

menor=b;

if(c<menor)

menor=c;

if(d<menor)

menor=d;

if(e<menor)

menor=e;

printf("O maior e %.2f e o menor e %.2f.",maior, menor);

}

1. Faça um programa que leia 3 nomes e os coloque em ordem alfabética.

char x[10],y[10],z[10];

main(){

printf("Diga tres nomes.\n");

scanf("%s%s%s",&x,&y,&z);

if((strcmp(x,y)<0)&&(strcmp(x,z)<0)&&(strcmp(y,z)<0))

printf("%s, %s, %s. Em ordem alfabetica.",x,y,z);

else

if((strcmp(x,y)<0)&&(strcmp(x,z)<0)&&(strcmp(z,y)<0))

printf("%s, %s, %s. Em ordem alfabetica.",x,z,y);

else

if((strcmp(y,x)<0)&&(strcmp(y,z)<0)&&(strcmp(x,z)<0))

printf("%s, %s, %s. Em ordem alfabetica.",y,x,z);

else

if((strcmp(y,x)<0)&&(strcmp(y,z)<0)&&(strcmp(z,x)<0))

printf("%s, %s, %s. Em ordem alfabetica.",y,z,x);

else

if((strcmp(z,x)<0)&&(strcmp(z,y)<0)&&(strcmp(x,y)<0))

printf("%s, %s, %s. Em ordem alfabetica.",z,x,y);

else

if((strcmp(z,x)<0)&&(strcmp(z,y)<0)&&(strcmp(y,x)<0))

printf("%s, %s, %s. Em ordem alfabetica.",z,y,x);

}

1. Faça um programa que leia 3 números e diga se eles podem ou não tomar um triângulo. Obs.: para formar um triângulo é necessário que a soma de dois lados seja sempre maior que o terceiro lado.

float lado1,lado2,lado3;

main(){

printf("Diga os lados do triangulo.\n");

scanf("%f%f%f",&lado1,&lado2,&lado3);

if(((lado1+lado2)>lado3)&&((lado2+lado3)>lado1)&&((lado1+lado3)>lado2))

printf("Esses lados podem formar um triangulo.");

else

printf("Nao e possivel formar um triangulo com esses lados.");

}

1. Faça um programa que leia 3 números e diga se eles podem ou não formar um triângulo, caso afirmativo, diga se o triângulo é equilátero, isósceles ou escaleno.

float lado1,lado2,lado3;

main(){

printf("Diga os lados do triangulo.\n");

scanf("%f%f%f",&lado1,&lado2,&lado3);

if(((lado1+lado2)>lado3)&&((lado2+lado3)>lado1)&&((lado1+lado3)>lado2)){

printf("Esses lados podem formar um triangulo.");

if((lado1==lado2)&&(lado1==lado3))

printf("E um triangulo equilatero.");

else

if((lado1!=lado2)&&(lado2!=lado3)&&(lado1!=lado3))

printf("E um triangulo escaleno.");

else

printf("E um triangulo isosceles.");

}

else

printf("Nao e possivel formar um triangulo com esses lados.");

}

1. Faça um programa que leia os coeficientes de uma equação do segundo grau e calcule as suas raízes. O programa deve mostrar quando possível o valor das raízes calculadas e a classificação das mesmas (raízes imaginárias, raiz única as raízes reais distintas).

float a,b,c,x;

main(){

printf("Diga os valores de a, b e c.\n");

scanf("%f%f%f",&a,&b,&c);

if(((b\*b)-4\*a\*c)<0)

printf("Raizes imaginarias.");

else

if(((b\*b)-4\*a\*c)==0){

c=-b/(2\*a);

printf("Raiz unica. %.2f",c);

}

else

if(((b\*b)-4\*a\*c)>0){

x=(-b+pow(((b\*b)-4\*a\*c),0.5))/(2\*a);

c=(-b-pow(((b\*b)-4\*a\*c),0.5))/(2\*a);

printf("Raizes reais distintas. %.2f e %.2f",c,x);

}

}

1. Um comerciante comprou um produto que quer vende-lo com um lucro de 45% se o valor da compra for menor que 20 BRL caso contrário o lucro será de 30%. Entre com o valor do produto e imprima o valor de venda.

float preco;

main(){

printf("Qual foi valor do produto?\n");

scanf("%f",&preco);

if(preco<20)

printf("O produto sera vendido por: %.2f.",preco\*1.45);

else

printf("O produto sera vendido por: %.2f.",preco\*1.3);

}

1. Faça um programa que leia nome, nota 1 e nota 2 de um aluno. Em seguida calcule a média e informe se o aluno está aprovado reprovado ou em prova final. A média maior ou igual a 7 é aprovação menor que 3 a reprovação e demais casos o aluno está em prova final.

float nota1,nota2;

main(){

printf("Diga as duas notas.\n");

scanf("%f%f",&nota1,&nota2);

nota1=(nota1+nota2)/2;

if(nota1>=7)

printf("Aprovado, media %.2f.",nota1);

else

if(nota1<3)

printf("Reprovado, media %.2f.",nota1);

else

printf("Prova final, media %.2f.",nota1);

}

1. Faça um programa que leia um número binário de 4 dígitos e diga quantos dígitos zero existem nesse número.

int b,contador;

main(){

printf("Diga o numero binario.\n");

scanf("%d",&b);

if(b/1000==0)

contador=contador+1;

if(((b/100)%10)==0)

contador=contador+1;

if(((b/10)%10)==0)

contador=contador+1;

if((b%10)==0)

contador=contador+1;

printf("Sao %d zeros.",contador);

}

## Estruturas Condicionais – Operador Ternário

1. Faça um programa que leia um número e diga se ele é par ou ímpar utilizando operador ternário.

int n;

main(){

printf("Diga o numero:\n");

scanf("%d",&n);

(n%2)?printf("%d e impar.",n):printf("%d e par",n);

}

1. Faça um programa que leia um número e diga se ele é positivo, negativo ou nulo utilizando operador ternário.

float n;

main(){

printf("Diga um numero:\n");

scanf("%f",&n);

(n==0)?printf("%d e nulo.",n):(n>0)?printf("%.2f e positivo",n):printf("%.2f e negativo.",n);

}

1. Faça um programa que leia três números e diga qual é o maior. Utilize operador ternário.

float x,y,z;

main(){

printf("Diga os tres numeros.\n");

scanf("%f%f%f",&x,&y,&z);

(x>y)&&(x>z)?printf("%.2f e o maior",x):(y>z)&&(y>x)?printf("%.2f e o maior",y):printf("%.2f e o maior",z);

}

char v;

main(){

printf("Diga uma vogal.\n");

scanf("%c",&v);

v=tolower(v);

(v=='a')||(v=='e')||(v=='i')||(v=='o')||(v=='u')?printf("%c e uma vogal",v):printf("%c nao e vogal.",v);

}

## Exercícios Condicionais 2

1. Dado três valores A, B e C, faça um programa que verifique se com esses valores é possível fazer um triângulo, e se for possível, diga se o triângulo retângulo = 90°, triângulo obtusângulo >90° ou triângulo acutângulo <90°.

float A,B,C,maior,menor,meio;

main(){

printf("Diga tres numeros para formar os lados do triangulo.\n");

scanf("%f%f%f",&A,&B,&C);

if((A>=(B+C))||(B>=(A+B))||(C>=(A+B)))

printf("Nao e possivel formar um triangulo com esses lados.");

else{

maior=A;

menor=A;

meio=A;

if((A!=B)&&(A!=C)&&(C!=B))

if(B>maior)

maior=B;

if(C>maior)

maior=C;

if(B<menor)

menor=B;

if(C<menor)

menor=C;

if((A!=maior)&&(A!=menor))

meio=A;

if((B!=maior)&&(B!=menor))

meio=B;

if((C!=maior)&&(C!=menor))

meio=C;

if((A==B)&&(A!=C))

if(C<A)

menor=C;

else

maior=C;

if((B==C)&&(B!=A))

if(B>A){

maior=B;

meio=C;

}

else{

meio=C;

menor=B;

}

if((C==A)&&(C!=B))

if(B<A)

menor=B;

else

maior=B;

if((maior\*maior)==((menor\*menor)+(meio\*meio)))

printf("Triangulo retangulo %.2f %.2f %.2f.",maior,meio,menor);

else

if((maior\*maior)<((menor\*menor)+(meio\*meio)))

printf("Triangulo acutangulo %.2f %.2f %.2f.",maior,meio,menor);

else

printf("Triangulo obtusangulo %.2f %.2f %.2f.",maior,meio,menor);

}

}

1. Dado três valores A, B e C, faça um programa que verifique se com esses valores é possível fazer um triângulo, e se um triângulo retângulo, determinar e imprimir seus ângulos internos.

float A,B,C;

main(){

printf("Diga tres numeros para formar os lados do triangulo.\n");

scanf("%f%f%f",&A,&B,&C);

if((A>(B+C))||(B>(A+B))||(C>(A+B)))

printf("Nao e possivel formar um triangulo com esses lados.");

else{

if((A>B)&&(A>C)&&(B>C))

if((A\*A)==((B\*B)+(C\*C)))

printf("O angulo entre %.2f e %.2f vale 90°, entre %.2f e %.2f vale 30° e entre %.2f e %.2f vale 60°. ",B,C,A,B,A,C);

else

printf("Nao e triangulo retangulo.");

if((A>B)&&(A>C)&&(C>B))

if((A\*A)==((B\*B)+(C\*C)))

printf("O angulo entre %.2f e %.2f vale 90°, entre %.2f e %.2f vale 30° e entre %.2f e %.2f vale 60°. ",B,C,A,C,A,B);

else

printf("Nao e triangulo retangulo.");

if((A>B)&&(A>C)&&(C==B))

if((A\*A)==((B\*B)+(C\*C)))

printf("O triangulo tem um angulo de 90° entre %.2f e %.2f, e dois outros angulos de 45°",B,C);

else

printf("Nao e triangulo retangulo.");

if((B>A)&&(B>C)&&(A>C))

if((B\*B)==((A\*A)+(C\*C)))

printf("O angulo entre %.2f e %.2f vale 90°, entre %.2f e %.2f vale 30° e entre %.2f e %.2f vale 60°. ",A,C,B,A,B,C);

else

printf("Nao e triangulo retangulo.");

if((B>A)&&(B>C)&&(C>A))

if((B\*B)==((A\*A)+(C\*C)))

printf("O angulo entre %.2f e %.2f vale 90°, entre %.2f e %.2f vale 30° e entre %.2f e %.2f vale 60°. ",A,C,B,C,B,A);

else

printf("Nao e triangulo retangulo.");

if((B>A)&&(B>C)&&(A==C))

if((B\*B)==((A\*A)+(C\*C)))

printf("O triangulo tem um angulo de 90° entre %.2f e %.2f, e dois outros angulos de 45°",A,C);

else

printf("Nao e triangulo retangulo.");

if((C>A)&&(C>B)&&(A>B))

if((C\*C)==((A\*A)+(B\*B)))

printf("O angulo entre %.2f e %.2f vale 90°, entre %.2f e %.2f vale 30° e entre %.2f e %.2f vale 60°. ",A,B,C,A,C,B);

else

printf("Nao e triangulo retangulo.");

if((C>A)&&(C>B)&&(B>A))

if((C\*C)==((A\*A)+(B\*B)))

printf("O angulo entre %.2f e %.2f vale 90°, entre %.2f e %.2f vale 30° e entre %.2f e %.2f vale 60°. ",A,B,C,B,C,A);

else

printf("Nao e triangulo retangulo.");

if((C>A)&&(C>B)&&(A==B))

if((C\*C)==((A\*A)+(B\*B)))

printf("O triangulo tem um angulo de 90° entre %.2f e %.2f, e dois outros angulos de 45°",A,B);

else

printf("Nao e triangulo retangulo.");

}

}

1. Faça um programa que leia um número x e calcule e imprima o valor de f(x).

int x;

main(){

printf("Diga o valor de x.\n");

scanf("%d",&x);

if(x<=1)

printf("f(x)=1");

else

if(x<=2)

printf("f(x)=2");

else

if(x<=3)

printf("f(x)=%d",x\*x);

else

printf("f(x)=%d",x\*x\*x);

}

1. Faça um programa que leia um número x e calcule e imprima o valor de f(x).

float x;

main(){

printf("Diga o valor de x.\n");

scanf("%f",&x);

x=((5\*x)+3)/(pow((pow(x,2)-16),0.5));

printf("f(x)=%.2f",x);

}

1. Faça um programa que a partir da idade e peso do paciente, calcule a dosagem de determinado medicamento e imprima a receita informando quantas gotas do medicamento o paciente deve tomar por dose. Considere que o medicamento em questão possui 500mg por ml, e que cada ml corresponde a 20 gotas.

int idade;

float peso, dosagem;

main(){

printf("Diga sua idade e seu peso.\n");

scanf("%d%f",&idade,&peso);

if((idade>=12)&&(peso>=60))

printf("40 gotas.");

else

if((idade>=12)&&(peso<60))

printf("35 gotas.");

else{

if((peso>=5)&&(peso<=9))

printf("5 gotas.");

if((peso>=9.1)&&(peso<=16))

printf("10 gotas.");

if((peso>=16.1)&&(peso<=24))

printf("15 gotas.");

if((peso>=24.1)&&(peso<=30))

printf("20 gotas.");

if(peso>30)

printf("30 gotas.");

}

}

1. Faça um programa que informe o total de calorias de uma refeição a partir do usuário que irá informar o prato, a sobremesa e a bebida. Veja a tabela a seguir. Mostre todas essas opções na tela.

int prato, sobremesa, bebida,calorias;

main(){

printf("Esses sao os pratos, sobremesas e bebidas disponiveis:\n================================================================================\n Pratos: 1- Vegetariano. 2- Peixe. 3- Frango. 4- Carne.\n Sobremesa: 1- Abacaxi. 2- Sorvete Diet. 3- Mouse Diet. 4- Mouse chocolate.\n Bebida: 1- Cha. 2- Suco de laranja. 3- Suco de melao. 4- Refrigerante Diet.\n================================================================================\nEscolha seu pedido, informe a numeracao na ordem Prato, Sobremesa, Bebida e diremos as calorias da refeicao.\n");

scanf("%d%d%d",&prato,&sobremesa,&bebida);

if(prato==1)

calorias=calorias+180;

if(prato==2)

calorias=calorias+230;

if(prato==3)

calorias=calorias+250;

if(prato==4)

calorias=calorias+350;

if(sobremesa==1)

calorias=calorias+75;

if(sobremesa==2)

calorias=calorias+110;

if(sobremesa==3)

calorias=calorias+170;

if(sobremesa==4)

calorias=calorias+200;

if(bebida==1)

calorias=calorias+20;

if(bebida==2)

calorias=calorias+70;

if(bebida==3)

calorias=calorias+100;

if(bebida==4)

calorias=calorias+65;

printf("Seu prato tem %d calorias.",calorias);

}

1. Faça um programa que leia o número correspondente ao mês atual e os dígitos (somente os quatro números) de uma placa de veículo, e através do número finalizador da placa (algarismo da casa das unidades) determine se o IPVA do veículo já venceu, vence no mês corrente ou ainda irá vencer.

int mes,placa;

main(){

printf("Diga a o mes atual e a placa do seu veiculo.\n");

scanf("%d%d",&mes,&placa);

placa=placa%10;

if(mes>=11)

printf("Ja venceu rapa.");

else{

if(placa==0){

if(mes==10)

printf("E nesse mes, corre.");

if(mes<=9)

printf("Ainda falta, ta safe.");

if(mes>10)

printf("Ja venceu rapa.");

}

else{

if(mes<placa)

printf("Ainda falta, ta safe.");

if(mes==placa)

printf("E nesse mes, corre.");

if(mes>placa)

printf("Ja venceu rapa.");

}

}

}

1. Faça um programa que verifique se um ano é bissexto. Anos divisíveis por 4 e não divisíveis por 100 são bissextos. Anos divisíveis por 400 são bissextos.

int ano;

main(){

printf("Diga qual e o ano.\n");

scanf("%d",&ano);

if(((ano%400)==0)||(((ano%4)==0)&&((ano%100)!=0)))

printf("E um ano bissexto.");

else

printf("Nao e um ano bissexto.");

}

1. Faça um programa que leia uma data no formato dd – mm – aaaa. E em seguida escreva qual será o próximo dia, mês e ano.

int dia,mes,ano;

main(){

printf("Diga qual e o dia, mes e ano, repectivamente.\n");

scanf("%d%d%d",&dia,&mes,&ano);

if((dia==31)&&(mes==12)){

dia=1;

mes=1;

ano=ano+1;

}

else

if((dia==28)&&(mes==2)&&(((ano%400)==0)||(((ano%4)==0)&&(ano%100)!=0))){

dia=29;

}

else

if(((dia==28)&&(mes==2)&&(!(((ano%400)==0)||(((ano%4)==0)&&(ano%100)!=0))))){

dia=1;

mes=3;

}

else{

if(((dia==31)&&((mes==1)||(mes==3)||(mes==5)||(mes==7)||(mes==8)||(mes==10)))||((dia==30)&&((mes==4)||(mes==6)||(mes==9)||(mes==11)))){

dia=1;

mes=mes+1;

}

else

dia=dia+1;

}

if((ano<1)||(mes>12)||(mes<1)||(dia>31)||(dia<1)||((dia==29)&&(mes==2)&&(!(((ano%400)==0)||(((ano%4)==0)&&(ano%100)!=0))))||((dia>=30)&&(mes==2))||((dia==31)&&((mes==4)||(mes==6)||(mes==9)||(mes==11))))

printf("Essa data nao existe seu pamonha.");

else

printf("Amanha e dia %d do mes %d do ano de %d.\n",dia,mes,ano);

}

1. Faça um programa que leia dois números binários de 3 dígitos e efetue a soma desses dois números binários.

int n1,n2,resultado,s;

main(){

printf("Diga os numeros binarios de tres digitos que serao somados.\n");

scanf("%d%d",&n1,&n2);

if(((n1%10)+(n2%10))==2){

resultado=0;

s=1;

}

else

if(((n1%10)+(n2%10))==1){

resultado=1;

s=0;

}

else{

resultado=0;

s=0;

}

if((((n1/10)%10)+((n2/10)%10)+s)==3){

resultado=resultado+10;

s=1;

}

else

if((((n1/10)%10)+((n2/10)%10)+s)==2)

s=1;

else

if((((n1/10)%10)+((n2/10)%10)+s)==1){

resultado=resultado+10;

s=0;

}

else{

s=0;

}

if(((n1/100)+(n2/100)+s)==3)

resultado=resultado+1100;

else

if(((n1/100)+(n2/100)+s)==2)

resultado=resultado+1000;

else

if(((n1/100)+(n2/100)+s)==1)

resultado=resultado+100;

if((n1>1000)||(n2>100))

printf("Esse numero tem mais de 3 casas decimais.");

else

printf("%d e o resultado da soma",resultado);

}

## Estruturas de Repetição – For

1. Faça um programa que imprima os números de 1 até 100 na tela de forma automática.

int x;

main(){

for(x=1;x<100;x++)

printf("%d, ",x);

}

1. Faça um programa que imprima os números de 100 até 1 na tela de forma automática.

int x;

main(){

for(x=100;x>0;x--)

printf("%d, ",x);

}

1. Preencha o que falta, no retângulo vermelho, para que o programa que imprima os números de 100 até 1 na tela de forma automática.

int i;

main(){

for(i=1;i<=100;i++)

printf("%d, ",101-i);

}

1. Faça um programa que mostre a tabuada de 1 até 10 de um número inteiro lido do teclado. O resultado deve aparecer da seguinte forma:

Se o número lido for 5.

1 x 5 = 5

2 x 5 = 10

.

.

.

10 x 5 = 50

int x,m;

main(){

printf("Diga o numero inteiro: ");

scanf("%d",&x);

if(x%1)

printf("Esse nao e um numero inteiro");

else

for(m=1;m<=10;m++)

printf("%d, ",m\*x);

}

1. Faça um programa que imprima os 100 primeiros números pares.

int n;

main(){

for(n=0;n<=100;n++)

printf("%d, ",n\*2);

}

1. Faça um programa que imprima os múltiplos de 5 no intervalo de 1 até 500.

int n;

main(){

for(n=1;n<=100;n++)

printf("%d, ",n\*5);

}

1. Faça um programa que imprima o quadrado dos números de 1 até 20.

int x;

main(){

for(x=1;x<=20;x++)

printf("%d, ",x\*x);

}

1. Faça um programa que imprima os números pares de 1 a 600.

int x;

main(){

for(x=1;x<=300;x++)

printf("%d, ",x\*2);

}

1. Faça um programa que leia 10 números e imprima a raiz quadrada de cada número.

int tempo;

float n;

main(){

for(tempo=1;tempo<=10;tempo++){

printf("Diga o numero: ");

scanf("%f",&n);

printf("%.2f\n",pow(n,0.5));

}

}

1. Faça um programa que leia 100 números inteiros e diga qual é o menor.

int n,menor,x;

main(){

printf("Diga o primeiro numero inteiro: ");

scanf("%d",&n);

menor=n;

for(x=1;x<=100;x++){

printf("Diga o proximo inteiro: ");

scanf("%d",&n);

if(n<menor)

menor=n;

}

printf("%d e o menor numero.",menor);

}

1. Faça um programa que leia 100 números inteiros e diga quantos são ímpares.

int x,impares,tempo;

main(){

impares=0;

for(tempo=1;tempo<=100;tempo++){

printf("Diga o numero: ");

scanf("%d",&x);

if((x%2)!=0)

impares=impares+1;

}

printf("Foram digitados %d numeros impares.",impares);

}

1. Faça um programa que leia 100 números e diga quantos estão abaixo de 1000, quantos são iguais a 1000 e quantos estão abaixo de 1000.

float n;

int tempo,acima,abaixo,mil;

main(){

acima=0;

abaixo=0;

mil=0;

for(tempo=1;tempo<=100;tempo++){

printf("Diga o numero: ");

scanf("%f",&n);

if(n==1000)

mil=mil+1;

if(n>1000)

acima=acima+1;

if(n<1000)

abaixo=abaixo+1;

}

printf("%d numeros acima de 1000 foram digitados, %d numeros foram menores, %d numeros sao iguais a mil.",acima,abaixo,mil);

}

1. Faça um programa que mostre na tela as letras de a até z.

char l;

main(){

for(l='a';l<='z';l++)

printf("%c, ",l);

}

1. Faça um programa que imprima a tabela ASCII.

char tempo;

main(){

for(tempo=33;tempo<=126;tempo++)

printf("%c, ",tempo);

}

1. Faça um programa que leia um número inteiro e calcule o seu fatorial.

int n,tempo;

main(){

printf("Diga o numero inteiro.\n");

scanf("%d",&n);

if(n==0)

n=1;

if(n<0)

printf("Esse numero e negativo.");

else{

for(tempo=n;tempo>=2;tempo--)

n=n\*(tempo-1);

printf("O fatorial do numero e %d",n);

}

}

1. Faça um programa que leia um número inteiro e calcule e mostre os n primeiros termos da série de Fibonacci.

int n,tempo,resposta,n1,n2;

main(){

n1=1;

n2=0;

printf("Quantos numeros da sequencia fibonacci serao apresentados?\n");

scanf("%d",&n);

for(tempo=0;tempo<=n;tempo++){

resposta=n1+n2;

printf("%d, ",resposta);

n2=n1;

n1=resposta;

}

}

1. Faça um programa que leia um número e verifique se ele é um número primo.

int tempo,x,denominadores;

main(){

printf("Diga o numero que voce quer saber se e primo.\n");

scanf("%d",&x);

if(x>1){

for(tempo=1;tempo<=x;tempo++)

if(x%tempo==0)

denominadores=denominadores+1;

if(denominadores>2)

printf("Nao e primo.");

else

printf("E primo.");

}

else

printf("Nao e primo.");

}

1. Um número perfeito é um número natural para o qual a soma de todos os seus divisores naturais próprios (excluindo ele mesmo) é igual ao próprio número. Exemplo 6 é um número perfeito, pois 6 = 1 + 2 +3. Exemplo 6 é um número perfeito, pois 6 = 1 + 2 +3. 28 é um número perfeito, pois 28 = 1+2+4+7+14. Faça um programa que leia um número e diga se ele é um número perfeito.

int tempo,x,divisores;

main(){

divisores=0;

printf("Diga o numero.\n");

scanf("%d",&x);

for(tempo=1;tempo<x;tempo++)

if(x%tempo==0)

divisores=divisores+tempo;

if(divisores==x)

printf("%d e um numero perfeito.",x);

else

printf("%d nao e perfeito.",x);

}

1. Faça um programa que leia um número inteiro n qualquer e mostre na tela a figura abaixo.

int n,tempo,tempoaux;

main(){

printf("Diga um numero.\n");

scanf("%d",&n);

for(tempo=1;tempo<=n;tempo++){

for(tempoaux=1;tempoaux<=tempo;tempoaux++)

printf("\*");

printf("\n");

}

}

1. Faça um programa que leia a altura e o sexo (codificado em 1=masculino e 2=feminino) de um grupo de n pessoas lidas do teclado.

Calcule e escreva:

a) Média da altura do grupo

b) Média da altura dos homens

c) Média da altura das mulheres

d) Maior altura do grupo

e) Menor altura do grupo

int tempo,n,sexo,m,f;

float altura,mediag,mediaf,mediam,maior,menor;

main(){

printf("Diga quantas pessoas fazem parte do grupo.\n");

scanf("%d",&n);

for(tempo=1;tempo<=n;tempo++){

printf("Diga o sexo e a altura, masculino e 1 e feminino e 2.\n");

scanf("%d%f",&sexo,&altura);

mediag=mediag+altura;

if(sexo==1){

mediam=mediam+altura;

m=m+1;

}

if(sexo==2){

mediaf=mediaf+altura;

f=f+1;

}

if(tempo==1){

menor=altura;

maior=altura;

}

if(altura>maior)

maior=altura;

if(altura<menor)

menor=altura;

}

if(m==0)

m=1;

if(f==0)

f=1;

printf("\nA media da altura do grupo e: %.2f,\nA media da altura das mulheres e %.2f.\nA media da altura dos homens e %.2f.\nA maior altura e %.2f e a menor e %.2f.",mediag/n,mediaf/f,mediam/m,maior,menor);

}

1. Faça um programa que leia um número inteiro n qualquer e mostre na tela a figura abaixo.

\*

\*\*

\*\*\*

\*\*

\*

int n, tempo,tempom,tempon;

main(){

printf("Qual e o tamanho do desenho?\n");

scanf("%d",&n);

for(tempo=1;tempo<=n;tempo++){

for(tempom=1;tempom<=tempo;tempom++)

printf("\*");

printf("\n");

}

for(tempo=n;tempo>=1;tempo--){

for(tempon=tempo-1;tempon>=1;tempon--)

printf("\*");

printf("\n");

}

}

## Estruturas de Repetição – While

1. Faça um programa que imprima os números de 1 até 100 na tela de forma automática.

int n;

main(){

n=1;

while(n<=100){

printf("%d, ",n);

n++;

}

}

1. Faça um programa que imprima os números de 100 até 1 na tela de forma automática.

int tempo;

main(){

tempo=100;

while(tempo>=1){

printf("%d, ",tempo);

tempo--;

}

}

1. Faça um programa que mostre a tabuada de 1 até 10 de um número inteiro lido do teclado. O resultado deve aparecer da seguinte forma:

Se o número lido for 5.

1 x 5 = 5

2 x 5 = 10

.

.

.

10 x 5 = 50

int tempo,n;

main(){

printf("Qual e o numero?\n");

scanf("%d",&n);

tempo=1;

while(tempo<=10){

printf("%d x %d = %d\n",tempo,n,n\*tempo);

tempo++;

}

}

1. Faça um programa que imprima os 100 primeiros números pares.

int tempo;

main(){

tempo=2;

while(tempo<=200){

printf("%d, ",tempo);

tempo+=2;

}

}

1. Faça um programa que imprima os múltiplos de 5 no intervalo de 1 até 500.

int tempo;

main(){

tempo=5;

while(tempo<=500){

printf("%d, ",tempo);

tempo+=5;

}

}

1. Faça um programa que imprima o quadrado dos números de 1 até 20.

int tempo;

main(){

tempo=1;

while(tempo<=20){

printf("%d, ",tempo\*tempo);

tempo++;

}

}

1. Faça um programa que imprima os números pares de 1 a 600.

int tempo;

main(){

tempo=2;

while(tempo<=600){

printf("%d, ",tempo);

tempo+=2;

}

}

1. Faça um programa que leia 10 números e imprima a raiz quadrada de cada número.

int tempo,x;

main(){

tempo=1;

while(tempo<=10){

printf("Diga um numero.\n");

scanf("%d",&x);

printf("A raiz quadrada desse numero e %.0f.\n",pow(x,0.5));

tempo++;

}

}

1. Faça um programa que leia 100 números inteiros e diga qual é o menor.

int tempo,x,menor;

main(){

tempo=1;

while(tempo<=100){

printf("Diga o numero.\n");

scanf("%d",&x);

if(tempo==1)

menor=x;

if(x<menor)

menor=x;

tempo++;

}

printf("%d e o menor dos numeros citados.",menor);

}

1. Faça um programa que leia 100 números inteiros e diga quantos são ímpares.

int tempo, i,x;

main(){

tempo=1;

while(tempo<=100){

printf("Diga um numero.\n");

scanf("%d",&x);

if(x%2)

i++;

tempo++;

}

printf("%d numeros ditos sao impares.",i);

}

1. Faça um programa que leia 100 números e diga quantos estão abaixo de 1000, quantos são iguais a 1000 e quantos estão abaixo de 1000.

int tempo,x,maiores,iguais,menores;

main(){

tempo=1;

while(tempo<=5){

printf("Diga o numero.\n");

scanf("%d",&x);

if(x==1000)

iguais++;

if(x>1000)

maiores++;

if(x<1000)

menores++;

tempo++;

}

printf("%d foram maiores que 1000, %d foram menores e %d foram iguais.",maiores,menores,iguais);

}

1. Faça um programa que mostre na tela as letras de a até z.

char tempo;

main(){

tempo='a';

while(tempo<='z'){

printf("%c, ",tempo);

tempo++;

}

}

1. Faça um programa que imprima a tabela ASCII.

char tempo;

main(){

tempo=33;

while(tempo<=126){

printf("%c, ",tempo);

tempo++;

}

}

1. Faça um programa que leia um número inteiro e calcule o seu fatorial.

int n,tempo;

main(){

printf("Diga o numero inteiro.\n");

scanf("%d",&n);

if(n==0)

n=1;

if(n<0)

printf("Esse numero e negativo.");

else{

tempo=n;

while(tempo>=2){

n\*=tempo-1;

tempo--;

}

printf("O fatorial do numero e %d",n);

}

}

1. Faça um programa que leia um número inteiro e calcule e mostre os n primeiros termos da série de Fibonacci.

int n,tempo,resposta,n1,n2;

main(){

n1=1;

n2=0;

printf("Quantos numeros da sequencia fibonacci serao apresentados?\n");

scanf("%d",&n);

tempo=0;

while(tempo<=n){

resposta=n1+n2;

printf("%d, ",resposta);

n2=n1;

n1=resposta;

tempo++

}

}

1. Faça um programa que leia um número e verifique se ele é um número primo.

int tempo,x,denominadores;

main(){

printf("Diga o numero que voce quer saber se e primo.\n");

scanf("%d",&x);

if(x>1){

tempo=1;

while(tempo<=x){

if(x%tempo==0)

denominadores=denominadores+1;

tempo++;

}

if(denominadores>2)

printf("Nao e primo.");

else

printf("E primo.");

}

else

printf("Nao e primo.");

}

1. Um número perfeito é um número natural para o qual a soma de todos os seus divisores naturais próprios (excluindo ele mesmo) é igual ao próprio número. Exemplo 6 é um número perfeito, pois 6 = 1 + 2 +3. Exemplo 6 é um número perfeito, pois 6 = 1 + 2 +3. 28 é um número perfeito, pois 28 = 1+2+4+7+14. Faça um programa que leia um número e diga se ele é um número perfeito.

int tempo,x,divisores;

main(){

divisores=0;

printf("Diga o numero.\n");

scanf("%d",&x);

tempo=1;

while(tempo<x){

if(x%tempo==0)

divisores=divisores+tempo;

tempo++;

}

if(divisores==x)

printf("%d e um numero perfeito.",x);

else

printf("%d nao e perfeito.",x);

}

1. Faça um programa que leia um número inteiro n qualquer e mostre na tela a figura abaixo.

int n,tempo,tempoaux;

main(){

printf("Diga um numero.\n");

scanf("%d",&n);

for(tempo=1;tempo<=n;tempo++){

for(tempoaux=1;tempoaux<=tempo;tempoaux++){

printf("\*");

if(tempoaux==tempo)

printf("\n");

}

}

}

1. Faça um programa que leia a altura e o sexo (codificado em 1=masculino e 2=feminino) de um grupo de n pessoas lidas do teclado.

Calcule e escreva:

a) Média da altura do grupo

b) Média da altura dos homens

c) Média da altura das mulheres

d) Maior altura do grupo

e) Menor altura do grupo

int tempo,n,sexo,m,f;

float altura,mediag,mediaf,mediam,maior,menor;

main(){

printf("Diga quantas pessoas fazem parte do grupo.\n");

scanf("%d",&n);

for(tempo=1;tempo<=n;tempo++){

printf("Diga o sexo e a altura, masculino e 1 e feminino e 2.\n");

scanf("%d%f",&sexo,&altura);

mediag=mediag+altura;

if(sexo==1){

mediam=mediam+altura;

m=m+1;

}

if(sexo==2){

mediaf=mediaf+altura;

f=f+1;

}

if(tempo==1){

menor=altura;

maior=altura;

}

if(altura>maior)

maior=altura;

if(altura<menor)

menor=altura;

}

if(m==0)

m=1;

if(f==0)

f=1;

printf("\nA media da altura do grupo e: %.2f,\nA media da altura das mulheres e %.2f.\nA media da altura dos homens e %.2f.\nA maior altura e %.2f e a menor e %.2f.",mediag/n,mediaf/f,mediam/m,maior,menor);

}

1. Faça um programa que calcule o fatorial de vários números. Quando for lido um número menor que zero, o programa termina.
2. Faça um programa que leia um número inteiro n e mostre os n primeiros números primos na tela.
3. Faça um programa que leia um número real x e um número inteiro y e calcule x elevado a y. Obs.: Não é permitido a utilização da função pow().

float x,r;

int y;

main(){

printf("Diga o numero e sua potencia.\n");

scanf("%f%d",&x,&y);

r=x;

while(y>=2){

r=r\*x;

y--;

}

printf("%.2f e a resposta.",r);

}

1. Faça um programa que leia um número binário qualquer e calcule o seu decimal correspondente.

int binario, decimal,ncasa,casa;

main(){

printf("Diga o numero binario.\n");

scanf("%d",&binario);

if(binario==0)

printf("Esse numero em decimal e %d.",binario);

else{

decimal=binario;

ncasa=0;

while(decimal!=0){

ncasa++;

decimal/=10;

}

decimal=0;

casa=0;

while(casa<=ncasa){

if(binario%10)

decimal=decimal+pow(2,casa);

binario/=10;

casa++;

}

printf("Esse numero em decimal e %d.",decimal);

}

}

1. Faça um programa que leia um número decimal qualquer e calcule o seu binário correspondente.

int decimal,binario,ncasa,casa,m;

main(){

printf("Diga o numero que sera convertido.\n");

scanf("%d",&decimal);

binario=decimal;

ncasa=1;

while(binario>=2){

binario/=2;

ncasa++;

}

binario=0;

casa=1;

m=1;

while(casa<=ncasa){

if(decimal%2)

binario=binario+1\*m;

m\*=10;

decimal/=2;

casa++;

}

printf("Esse numero em binario e %d.",binario);

}

1. Diga quantas vezes a senteça S será executada.

a)

b=0;

do{

c=1 && 0;

S;

b=!b;

}while(!(!b)&&(!c));

2 vezes

/\*

b c s

0 0 1

1 0 1

0 1 1

1 0 1

0 0 2

1 1 2

\*/

b)

b=0;

do{

while (b)

S;

b=!b;

}while(b);

1 vez só.

/\* b s

0 0

1 1

0

\*/

c)

i=0;

for(j=1; j<=i; j++)

for(l=1; l<=j; l++)

S;

0 vezes, porque j, que vale 1 é maior que i, que vale 0, então o programa pula o for.

1. Reescreva o trecho abaixo utilizando while e do while.

i=0;

for(j=1; j<=i; j++)

for(l=1; l<=j; l++)

S;

i=0;

j=1;

l=1;

while(j<=i){

while(l<=){

S;

l++;

}

j++;

}

i=0;

j=1;

l=1;

do{

do{

S;

l++;

}(l<=j);

j++;

}while(j<=i);

1. Faça um programa que calcule o valor de S(n).
2. S(n) = x/1 – x/2 + x/3 – x/4 + x/5 - .... +- x/n

int tempo,x,Sn,aux;

main(){

printf("Quantas vezes sera repetido?\n");

scanf("%d",&Sn);

printf("Qual e o valor de x?\n");

scanf("%d",&x);

aux=Sn;

Sn=0;

for(tempo=1;tempo<=aux;tempo++){

if(tempo%2)

Sn=Sn+(x/tempo);

else

Sn=Sn-(x/tempo);

}

printf("%d.",Sn);

}

1. S(n) = 2/1 + 3/1 + 5/2 + 7/3 + 11/5 + 13/8 + 17/13 + 19/21 + ... (n primeiros termos)

int Snaux,tempo,n1,n2,fibotempo,primo,denominadores,tempoprimo;

float Sn,fibo;

main(){

printf("Qual e o valor de Sn?\n");

scanf("%d",&Snaux);

Sn=0;

n1=1;

n2=0;

primo=1;

tempo=1;

fibo=1;

while(tempo<=Snaux){

for(fibotempo=1;fibotempo<=tempo;fibotempo++){

if(fibotempo==1){

fibo=1;

n2=1;

n1=1;

}

else

if(fibotempo==2){

fibo=1;

n2=1;

n1=1;

}

else{

fibo=n1+n2;

n2=n1;

n1=fibo;

}

}

while(denominadores!=2){

denominadores=0;

primo++;

for(tempoprimo=1;tempoprimo<=primo;tempoprimo++)

if((primo%tempoprimo)==0)

denominadores++;

}

denominadores=0;

Sn=Sn+(primo/fibo);

printf("%.2f=%d / %.2f\n\n",Sn,primo,fibo);

tempo++;

}

printf("%.2f.",Sn);

}

## Estruturas de Repetição – do while

1) Faça um programa que imprima os números de 1 até 100 na tela de forma automática.

int n;

main(){

n=1;

do{

printf("%d, ",n);

n++;

}while(n<=100);

}

2) Faça um programa que imprima os números de 100 até 1 na tela de forma automática.

int n;

main(){

n=100;

do{

printf("%d, ",n);

n--;

}while(n>=1);

}

4) Faça um programa que mostre a tabuada de 1 até 10 de um número inteiro lido do teclado. O resultado deve aparecer da seguinte forma:

Se o número lido for 5.

1 x 5 = 5

2 x 5 = 10

.

.

.

10 x 5 = 50

int tempo,x;

main(){

printf("Diga o numero.\n");

scanf("%d",&x);

tempo=1;

do{

printf("%d x %d = %d\n",tempo,x,x\*tempo);

tempo++;

}while(tempo<=10);

}

5) Faça um programa que imprima os 100 primeiros números pares.

int tempo;

main(){

tempo=2;

do{

printf("%d, ",tempo);

tempo+=2;

}while(tempo<=200);

}

6) Faça um programa que imprima os múltiplos de 5 no intervalo de 1 até 500.

int tempo;

main(){

tempo=5;

do{

printf("%d, ",tempo);

tempo+=5;

}while(tempo<=500);

}

7) Faça um programa que imprima o quadrado dos números de 1 até 20.

int tempo;

main(){

tempo=1;

do{

printf("%d, ",tempo\*tempo);

tempo++;

}while(tempo<=20);

}

8) Faça um programa que imprima os números pares de 1 a 600.

int tempo;

main(){

tempo=2;

do{

printf("%d, ",tempo);

tempo+=2;

}while(tempo<=600);

}

9) Faça um programa que leia 10 números e imprima a raiz quadrada de cada número.

int tempo;

float x;

main(){

tempo=1;

do{

printf("Diga o numero que voce quer saber a raiz.\n");

scanf("%f",&x);

printf("%.2f e a raiz.\n\n",pow(x,0.5));

tempo++;

}while(tempo<=3);

}

10) Faça um programa que leia 100 números inteiros e diga qual é o menor.

int n,menor,x;

main(){

printf("Diga o primeiro numero inteiro: ");

scanf("%d",&n);

menor=n;

for(x=1;x<=100;x++){

printf("Diga o proximo inteiro: ");

scanf("%d",&n);

if(n<menor)

menor=n;

}

printf("%d e o menor numero.",menor);

}

11) Faça um programa que leia 100 números inteiros e diga quantos são ímpares.

int x,impares,tempo;

main(){

impares=0;

for(tempo=1;tempo<=100;tempo++){

printf("Diga o numero: ");

scanf("%d",&x);

if((x%2)!=0)

impares=impares+1;

}

printf("Foram digitados %d numeros impares.",impares);

}

12) Faça um programa que leia 100 números e diga quantos estão abaixo de 1000, quantos são iguais a 1000 e quantos estão abaixo de 1000.

float n;

int tempo,acima,abaixo,mil;

main(){

acima=0;

abaixo=0;

mil=0;

for(tempo=1;tempo<=100;tempo++){

printf("Diga o numero: ");

scanf("%f",&n);

if(n==1000)

mil=mil+1;

if(n>1000)

acima=acima+1;

if(n<1000)

abaixo=abaixo+1;

}

printf("%d numeros acima de 1000 foram digitados, %d numeros foram menores, %d numeros sao iguais a mil.",acima,abaixo,mil);

}

13) Faça um programa que mostre na tela as letras de a até z.

char l;

main(){

for(l='a';l<='z';l++)

printf("%c, ",l);

}

14) Faça um programa que imprima a tabela ASCII.

char tempo;

main(){

for(tempo=33;tempo<=126;tempo++)

printf("%c, ",tempo);

}

15) Faça um programa que leia um número inteiro e calcule o seu fatorial.

int n,tempo;

main(){

printf("Diga o numero inteiro.\n");

scanf("%d",&n);

if(n==0)

n=1;

if(n<0)

printf("Esse numero e negativo.");

else{

for(tempo=n;tempo>=2;tempo--)

n=n\*(tempo-1);

printf("O fatorial do numero e %d",n);

}

}

16) Faça um programa que leia um número inteiro e calcule e mostre os n primeiros termos da série de Fibonacci.

int n,tempo,resposta,n1,n2;

main(){

n1=1;

n2=0;

printf("Quantos numeros da sequencia fibonacci serao apresentados?\n");

scanf("%d",&n);

for(tempo=0;tempo<=n;tempo++){

resposta=n1+n2;

printf("%d, ",resposta);

n2=n1;

n1=resposta;

}

}

17) Faça um programa que leia um número e verifique se ele é um número primo.

int tempo,x,denominadores;

main(){

printf("Diga o numero que voce quer saber se e primo.\n");

scanf("%d",&x);

if(x>1){

for(tempo=1;tempo<=x;tempo++)

if(x%tempo==0)

denominadores=denominadores+1;

if(denominadores>2)

printf("Nao e primo.");

else

printf("E primo.");

}

else

printf("Nao e primo.");

}

18) Um número perfeito é um número natural para o qual a soma de todos os seus divisores naturais próprios (excluindo ele mesmo) é igual ao próprio número. Exemplo 6 é um número perfeito, pois 6 = 1 + 2 +3. Exemplo 6 é um número perfeito, pois 6 = 1 + 2 +3. 28 é um número perfeito, pois 28 = 1+2+4+7+14. Faça um programa que leia um número e diga se ele é um número perfeito.

int tempo,x,divisores;

main(){

divisores=0;

printf("Diga o numero.\n");

scanf("%d",&x);

for(tempo=1;tempo<x;tempo++)

if(x%tempo==0)

divisores=divisores+tempo;

if(divisores==x)

printf("%d e um numero perfeito.",x);

else

printf("%d nao e perfeito.",x);

}

19) Faça um programa que leia um número inteiro n qualquer e mostre na tela a figura abaixo.

int n,tempo,tempoaux;

main(){

printf("Diga um numero.\n");

scanf("%d",&n);

for(tempo=1;tempo<=n;tempo++){

for(tempoaux=1;tempoaux<=tempo;tempoaux++){

printf("\*");

if(tempoaux==tempo)

printf("\n");

}

}

}

20) Faça um programa que leia a altura e o sexo (codificado em 1=masculino e 2=feminino) de um grupo de n pessoas lidas do teclado.

Calcule e escreva:

a) Média da altura do grupo

b) Média da altura dos homens

c) Média da altura das mulheres

d) Maior altura do grupo

e) Menor altura do grupo

int tempo,n,sexo,m,f;

float altura,mediag,mediaf,mediam,maior,menor;

main(){

printf("Diga quantas pessoas fazem parte do grupo.\n");

scanf("%d",&n);

for(tempo=1;tempo<=n;tempo++){

printf("Diga o sexo e a altura, masculino e 1 e feminino e 2.\n");

scanf("%d%f",&sexo,&altura);

mediag=mediag+altura;

if(sexo==1){

mediam=mediam+altura;

m=m+1;

}

if(sexo==2){

mediaf=mediaf+altura;

f=f+1;

}

if(tempo==1){

menor=altura;

maior=altura;

}

if(altura>maior)

maior=altura;

if(altura<menor)

menor=altura;

}

if(m==0)

m=1;

if(f==0)

f=1;

printf("\nA media da altura do grupo e: %.2f,\nA media da altura das mulheres e %.2f.\nA media da altura dos homens e %.2f.\nA maior altura e %.2f e a menor e %.2f.",mediag/n,mediaf/f,mediam/m,maior,menor);

}

## Estrutura de Dados Homogêneos – Vetor

1. Dado um vetor de 100 elementos, determine o maior e o menor elemento do vetor

#define T 100

int tempo;

float maior,menor,x[T];

main(){

for(tempo=0;tempo<T;tempo++){

printf("Diga o %d° numero.\n",tempo+1);

scanf("%f",&x[tempo]);

if(tempo==0){

maior=x[tempo];

menor=x[tempo];

}

else{

if(x[tempo]>maior)

maior=x[tempo];

if(x[tempo]<menor)

menor=x[tempo];

}

}

printf("%.2f e o menor numero e %.2f e o maior.",menor,maior);

}

1. Dado um vetor de 100 elementos, determine a diferença entre a soma dos elementos de índice par e a soma dos elementos de índice ímpar.

#define TAM 100

int tempo,valor[TAM],pares,impares,diferenca;

main(){

pares=0;

impares=0;

diferenca=0;

for(tempo=0;tempo<TAM;tempo++){

printf("insira o %d valor.\n",tempo+1);

scanf("%d",&valor[tempo]);

if(valor[tempo]%2)

impares+=valor[tempo];

else

pares+=valor[tempo];

}

if(pares>impares)

diferenca=pares-impares;

else

diferenca=impares-pares;

printf("A diferenca entre os pares e impares e %d.",diferenca);

}

1. Dado um vetor com 20 elementos, gerar outro vetor que contenha somente números múltiplos de 3 encontrados no primeiro vetor.

#define TAM 20

int tempo,valor[TAM],m[TAM],tempom;

main(){

tempom=0;

for(tempo=0;tempo<TAM;tempo++){

printf("Entre com o valor.\n");

scanf("%d",&valor[tempo]);

if((valor[tempo]%3)==0){

m[tempom]=valor[tempo];

tempom++;

}

}

for(tempo=0;tempo<tempom;tempo++)

printf("%d, ",m[tempo]);

}

1. Existem 20 candidatos à presidência, com códigos que variam de 1 até 20. Codificou-se 21 para votos brancos e 22 para votos nulos. Cada voto vem em um cartão (contendo só um voto) e o último cartão vem com o número 0 (zero). Faça um programa que auxilie a operação dos votos, imprimindo a quantidade de votos de cada candidato, o número de votos em branco, o número de votos nulos e o total de votantes. Imprima também o(s) candidato(s) que venceram a eleição e o número de votos do(s) vencedor(es).

#define TAM 23

int vencedor,votosvencedor,candidato[TAM],gados,tempo,empate;

main(){

gados=-1;

for(tempo=0;tempo<TAM;tempo++)

candidato[tempo]=0;

tempo=0;

empate=0;

printf("Escreva o numero 0 para encerrar a votacao.\n");

while(candidato[0]==0){

printf("Diga o numero do seu candidato.\n");

scanf("%d",&tempo);

if(tempo>=0&&tempo<23){

candidato[tempo]++;

gados++;

}

else

printf("ENTRE COM UM VALOR VALIDO.\n");

}

for(tempo=0;tempo<TAM-2;tempo++){

if(tempo==0){

votosvencedor=candidato[tempo];

vencedor=0;

}

else

if(candidato[tempo]==votosvencedor){

empate++;

}

else

if(candidato[tempo]>votosvencedor){

vencedor=tempo;

votosvencedor=candidato[tempo];

}

}

if(empate==0)

printf("%d gados votaram, o proximo ditador sera o cadidato numero %d, que ganhou com %d votos. Agora diremos a quantidade de votos de cada um e, em seguida os votos brancos e nulos.\n",gados,vencedor,votosvencedor);

else{

printf("Houve um empate nas eleicoes, os seguintes candidatos empataram, com %d votos:\n",votosvencedor);

for(tempo=1;tempo<TAM-2;tempo++){

if(candidato[tempo]==votosvencedor)

printf("%d\n",tempo);

}

}

for(tempo=1;tempo<TAM;tempo++){

if(tempo==21)

printf("Houveram %d votos brancos.\n",candidato[tempo]);

else

if(tempo==22)

printf("Houveram %d votos nulos.\n",candidato[tempo]);

else

printf("O candidato %d teve %d votos.\n",tempo,candidato[tempo]);

}

}

1. Um palíndromo é uma sequência de caracteres cuja leitura é idêntica se feita da direita para a esquerda e vice versa. Exemplo: osso e ovo são palíndromos. Nas expressões mais extensas os espaços são ignorados. Ex.: subi no onibus. Faça um programa que leia uma sequência de caracteres terminada pelo caractere ponto final, mostre-a e diga se é palíndroma ou não. Obs.: a sequência não possui mais que 50 caracteres. Só será permitida a utilização de um único vetor.

int inicio, fim,tamanho,tempo,p;

char letra[50],letra\_analisada;

main(){

tamanho=0;

printf("Diga uma frase e termine com ponto final.\n");

scanf("%c",&letra\_analisada);

while(letra\_analisada!='.'){

if(letra\_analisada!=' '){

letra[tamanho]=letra\_analisada;

tamanho++;

}

scanf("%c",&letra\_analisada);

}

tempo=1;

inicio=0;

fim=tamanho-1;

while(tempo<=(tamanho/2)){

printf("%c e %c\n",letra[inicio],letra[fim]);

if(letra[inicio]!=letra[fim])

p++;

inicio++;

fim--;

tempo++;

}

if(p>0)

printf("A frase nao e palindroma.");

else

printf("A frase e palindroma.");

}

## Estruturas de dados homogêneos - Matriz

1. Faça um programa que leia uma matriz 100x100, de números inteiros, e mostre qual é o maior e o menor elemento dessa matriz.

#define X 100

#define Y 100

int matriz[X][Y],tempox,tempoy,maior,menor;

main(){

for(tempox=0;tempox<X;tempox++)

for(tempoy=0;tempoy<Y;tempoy++){

printf("Insira o proximo numero.\n");

scanf("%d",&matriz[tempox][tempoy]);

if(tempox==0&&tempoy==0)

maior=matriz[tempox][tempoy];

menor=matriz[tempox][tempoy];

if(matriz[tempox][tempoy]>maior)

maior=matriz[tempox][tempoy];

if(matriz[tempox][tempoy]<menor)

menor=matriz[tempox][tempoy];

}

printf("%d e o maior numero citado e %d e o menor.",maior,menor);

}

1. Faça um programa que preencha os elementos de uma matriz 5x5, de números inteiros, e em seguida calcule a soma de todos os elementos da 3ª linha com os elementos da 5ª linha, o produto dos elementos da 4ª coluna, o somatório dos elementos do diagonal principal e o somatório dos elementos da diagonal secundária.

#define X 5

#define Y 5

int matriz[X][Y],tempox,tempoy,soma,produto,diagp,diags;

main(){

soma=0;

produto=1;

diagp=0;

diags=0;

for(tempox=0;tempox<X;tempox++){

for(tempoy=0;tempoy<Y;tempoy++){

printf("Diga o proximo numero.\n");

scanf("%d",&matriz[tempox][tempoy]);

if(tempox==2||tempox==4)

soma+=matriz[tempox][tempoy];

if(tempoy==3)

produto\*=matriz[tempox][tempoy];

if(tempox==tempoy)

diagp+=matriz[tempox][tempoy];

if(tempoy==(X-1-tempox))

diags+=matriz[tempox][tempoy];

}

}

printf("A soma dos numeros da 3 e 5 linhas e %d, o produto dos da 4 e %d, a soma da diagonal principal e %d e da secundaria e %d.",soma,produto,diagp,diags);

}

1. Faça um programa que preencha os elementos de uma matriz 50x50 e em seguida calcule a soma dessas duas matrizes em uma outra matriz e mostre o resultado dessa matriz soma na tela.

#define X 2

#define Y 2

int tempox,tempoy,matriz1[X][Y],matriz2[X][Y],soma[X][Y];

main(){

for(tempox=0;tempox<X;tempox++)

for(tempoy=0;tempoy<Y;tempoy++){

soma[tempox][tempoy]=0;

}

for(tempox=0;tempox<X;tempox++)

for(tempoy=0;tempoy<Y;tempoy++){

printf("Insira o proximo numero.\n");

scanf("%d",&matriz1[tempox][tempoy]);

soma[tempox][tempoy]+=matriz1[tempox][tempoy];

}

for(tempox=0;tempox<X;tempox++)

for(tempoy=0;tempoy<Y;tempoy++){

printf("Insira o proximo numero.\n");

scanf("%d",&matriz2[tempox][tempoy]);

soma[tempox][tempoy]+=matriz2[tempox][tempoy];

}

for(tempox=0;tempox<X;tempox++){

for(tempoy=0;tempoy<Y;tempoy++)

printf("%d, ",soma[tempox][tempoy]);

printf("\n");

}

}

1. Na teoria de sistemas, define-se como elemento minimax de uma matriz o menor elemento da linha onde se encontra o maior elemento da matriz. Faça um programa que leia uma matriz 100x100 de números inteiros e encontre seu elemento minimax, mostrando também a sua posição.

#define X 100

#define Y 100

int matriz[X][Y],tempox,tempoy,maior,minimax,linhamaior;

main(){

for(tempox=0;tempox<X;tempox++)

for(tempoy=0;tempoy<Y;tempoy++){

printf("Diga o proximo numero.\n");

scanf("%d",&matriz[tempox][tempoy]);

if(tempox==0&&tempoy==0)

maior=matriz[tempox][tempoy];

if(matriz[tempox][tempoy]>maior){

maior=matriz[tempox][tempoy];

minimax=matriz[tempox][tempoy];

linhamaior=tempox;

}

}

for(tempoy=0;tempoy<Y;tempoy++)

if(matriz[linhamaior][tempoy]<minimax)

minimax=matriz[linhamaior][tempoy];

printf("%d e o minimax.",minimax);

}

1. Faça um programa que leia uma matriz 50x50, e imprima o número de linhas e o número de colunas nulas da matriz. Exemplo: matriz 4x4 essa matriz possui 2 linhas nulas e 1 coluna nula.

#define X 50

#define Y 50

int matriz[X][Y],tempox,tempoy,linhas,colunas,zeros;

main(){

for(tempox=0;tempox<X;tempox++){

zeros=0;

for(tempoy=0;tempoy<Y;tempoy++){

printf("Insira o proximo valor.\n");

scanf("%d",&matriz[tempox][tempoy]);

if(matriz[tempox][tempoy]==0)

zeros++;

if(zeros==Y)

linhas++;

}

}

for(tempoy=0;tempoy<Y;tempoy++){

zeros=0;

for(tempox=0;tempox<X;tempox++){

if(matriz[tempox][tempoy]==0)

zeros++;

if(zeros==X)

colunas++;

}

}

printf("%d colunas e %d linhas sao nulas.",colunas,linhas);

}

1. Um quadrado mágico, é uma matriz quadrada, cuja soma de cada linha, coluna ou diagonal é a mesma. Faça um programa que leia uma matriz 50x50, e diga se essa matriz é um quadrado mágico. Exemplo de um quadrado mágico 4x4. A soma de qualquer linha, coluna ou diagonal da sempre 34.

#define TAM 50

int matriz[TAM][TAM],slinha[TAM],scoluna[TAM],sdp,sds,tempoy,tempox,falso;

main(){

for(tempox=0;tempox<TAM;tempox++){

slinha[tempox]=0;

scoluna[tempox]=0;

}

for(tempoy=0;tempoy<TAM;tempoy++)

for(tempox=0;tempox<TAM;tempox++){

printf("Insira o proximo numero.\n");

scanf("%d",&matriz[tempoy][tempox]);

slinha[tempox]+=matriz[tempoy][tempox];

scoluna[tempoy]+=matriz[tempoy][tempox];

if(tempox==tempoy)

sdp+=matriz[tempoy][tempox];

if(tempox==(TAM-1-tempoy))

sds+=matriz[tempoy][tempox];

}

for(tempox=1;tempox<TAM;tempox++){

if(slinha[tempox]!=slinha[tempox-1])

falso++;

if(scoluna[tempox]!=scoluna[tempox-1])

falso++;

}

if(sdp!=sds){

falso++;

printf("%d e diferente de %d.",sdp,sds);

}

if(falso>0)

printf("Nao e um quadrado magico.");

else

printf("E um quadrado magico.");

}

1. Uma imagem colorida RGB (Red Green Blue) é composta por 3 matrizes, uma para o canal Red, uma para o canal Green e outra para o canal Blue, que podem assumir valores de 0 a 255. Uma imagem com tons de cinza pode ser obtida através da média aritmética das matrizes RGB. Faça um programa que leia uma imagem colorida de tamanho 500x700, ou seja as matrizes RGB e calcule a sua respectiva matriz em tons de cinza.

#define X 500

#define Y 700

int red[Y][X],green[Y][X],blue[Y][X],gray[Y][X],tempox,tempoy;

main(){

for(tempoy=0;tempoy<Y;tempoy++)

for(tempox=0;tempox<X;tempox++){

printf("Diga o valor RGB do pixel na posicao[%d][%d].\n",tempoy,tempox);

scanf("%d%d%d",&red[tempoy][tempox],&green[tempoy][tempox],&blue[tempoy][tempox]);

while(red[tempoy][tempox]<0||red[tempoy][tempox]>255||green[tempoy][tempox]<0||green[tempoy][tempox]>255||blue[tempoy][tempox]<0||blue[tempoy][tempox]>255){

printf("Digite valores entre 0 e 255 para posicao[%d][%d]",tempoy,tempox);

scanf("%d%d%d",&red[tempoy][tempox],&green[tempoy][tempox],&blue[tempoy][tempox]);

}

gray[tempoy][tempox]=red[tempoy][tempox]+green[tempoy][tempox]+blue[tempoy][tempox];

}

printf("Sua matriz em tons de cinza é:\n");

for(tempoy=0;tempoy<Y;tempoy++){

for(tempox=0;tempox<X;tempox++)

printf("%d ",gray[tempoy][tempox]/3);

printf("\n");

}

}

1. Faça um programa que leia uma imagem em tons de cinza de tamanho 500x700 e calcule o seu histograma. O histograma calcula a quantidade de cada tom de pixel na imagem (frequência de pixels em uma imagem). Cada pixel pode assumir um valor entre 0 a 255.

#define X 2

#define Y 2

int gray[Y][X],numero[256],tempox,tempoy;

main(){

for(tempox=0;tempox<X;tempox++)

numero[tempox]=0;

for(tempoy=0;tempoy<Y;tempoy++)

for(tempox=0;tempox<X;tempox++){

printf("Diga o proximo valor.\n");

scanf("%d",&gray[tempoy][tempox]);

while(gray[tempoy][tempox]>255||gray[tempoy][tempox]<0){

printf("Diga um valor valido.\n");

scanf("%d",gray[tempoy][tempox]);

}

numero[gray[tempoy][tempox]]++;

}

for(tempox=0;tempox<255;tempox++)

if(numero[tempox]>0)

printf("%d apareceu %d vezes.\n",tempox,numero[tempox]);

}

## Modularização - Função

1. Faça uma função que receba um número como parâmetro e retorne 1 se o número for par e 0 se o número for ímpar.

int numero;

int evenodd(int numero);

main(){

printf("Diga o numero.\n");

scanf("%d",&numero);

if(evenodd(numero))

printf("%d e par.",numero);

else

printf("%d e impar.",numero);

}

int evenodd(int numero){

if(numero%2)

return 0;

else

return 1;

}

1. Faça uma função que receba um caractere alfa numérico e retorne 1 se for vogal e zero se não for.

char caracter;

int vogal(char valor);

main(){

printf("Diga o caracter.\n");

scanf("%c",&caracter);

if(vogal(caracter))

printf("%c e uma vogal.",caracter);

else

printf("%c nao e uma vogal.",caracter);

}

int vogal(valor){

valor=toupper(char valor);

if(valor=='A'||valor=='E'||valor=='I'||valor=='O'||valor=='U')

return 1;

else

return 0;

}

1. Faça uma função que receba um caractere alfa numérico e retorne 1 se for consoante e zero se não for.

char caracter;

int consoante(char valor);

main(){

printf("Diga o caracter alfanumerico.\n");

scanf("%c",&caracter);

if(consoante(caracter))

printf("%c e consoante.",caracter);

else

printf("%c nao e consoante.",caracter);

}

int consoante(char valor){

valor=tolower(valor);

if((valor>'a'&&valor<'z')&&(valor!='a'||valor!='e'||valor!='i'||valor!='o'||valor!='u'))

return 1;

else

return 0;

}

1. Faça uma função que receba um número inteiro como parâmetro e retorne o fatorial desse número

int numero;

int fatorial(int num);

main(){

printf("Diga o numero.\n");

scanf("%d",&numero);

printf("O fatorial de %d e %d.",numero,fatorial(numero));

}

fatorial(int num){

int tempo;

for(tempo=num-1;tempo>0;tempo--)

num\*=tempo;

return num;

}

1. Faça as funções necessárias para que o programa abaixo funcione corretamente.

float a, b, c;

float delta(float a,float b,float c);

float raiz1(float a,float b,float delta);

float raiz2(float a,float b,float delta);

main(){

printf("Entre com os coeficientes da equação do segundo grau:");

scanf("%f%f%f",&a,&b,&c);

if (delta(a,b,c)<0)

printf("Raízes imaginárias %d");

else

if (delta(a,b,c)==0)

printf("Raiz única = %.2f", raiz1(a, b, delta(a,b,c) ) );

else

printf("Raizes distintas. x1 = %.2f e x2=%.2f ", raiz1(a, b, delta(a,b,c) ), raiz2(a, b, delta(a,b,c) ));

}

float delta(float a,float b,float c){

return ((b\*b)-(4\*a\*c));

}

float raiz1(float a,float b,float delta){

return (((-b+pow(delta,0.5)))/(2\*a));

}

float raiz2(float a,float b,float delta){

return (((-b-pow(delta,0.5)))/(2\*a));

}

1. Faça uma função que receba como parâmetro um número inteiro e retorne o reverso desse número, por exemplo 1567 =>

int numero;

int numero\_reverso(int num);

main(){

printf("Diga um numero inteiro.\n");

scanf("%d",&numero);

printf("%d ao contrario e %d.",numero,numero\_reverso(numero));

}

int numero\_reverso(int num){

int reverso,casa,casas;

reverso=num;

while(reverso>0){

reverso/=10;

casas++;

}

reverso=0;

for(casa=casas-1;casa>-1;casa--){

reverso+=(num%10)\*(pow(10,casa));

num/=10;

}

return reverso;

}

1. Faça uma função que receba como parâmetro a altura e o sexo de uma pessoa (0 – masculino e 1 – feminino), em seguida a função deve retornar o peso ideal dessa pessoa. Homens => (72,7\*altura) - 58 e mulheres => (62,1\*altura)-44,7.

int genero;

float altura;

float peso\_ideal(int gerero, float altura);

main(){

printf("Diga o genero (0 para homem e 1 para mulher) e sua altura.\n");

scanf("%d%f",&genero,&altura);

while(altura<0||(genero!=0&&genero!=1)){

printf("Insira valores validos.\n");

scanf("%d%f",&genero,&altura);

}

printf("Seu peso ideal e %.2f",peso\_ideal(genero,altura));

}

float peso\_ideal(int genero,float altura){

if(genero==0)

return ((72.7\*altura)-58);

if(genero==1)

return ((62.1\*altura)-44.7);

}

8. Faça um programa que leia a altura e sexo de um grupo de pessoas, e imprima o peso ideal até que seja digitado um valor negativo para a altura. Utilize a função criada no exercício 7 para calcular o peso ideal.

int genero;

float altura;

float peso\_ideal(int gerero, float altura);

main(){

altura=1;

while(altura!=0){

printf("Diga o genero (0 para homem e 1 para mulher) e sua altura.\n");

scanf("%d%f",&genero,&altura);

while(genero!=0&&genero!=1){

printf("Insira valores validos.\n");

scanf("%d%f",&genero,&altura);

}

if(altura>0)

printf("Seu peso ideal e %.2f.\n\n",peso\_ideal(genero,altura));

}

}

float peso\_ideal(int genero,float altura){

if(genero==0)

return ((72.7\*altura)-58);

if(genero==1)

return ((62.1\*altura)-44.7);

}

9. Um número de Armstrong é um número inteiro constituído de n dígitos que tem a seguinte propriedade: seu valor é igual à soma de cada um dos valores de seus dígitos componentes elevado à n-ésima potência. Por exemplo, o número 153 (constituído de 3 dígitos) é igual a 13+53+33. Portanto 153 é um número de Armstrog. Abaixo está sendo mostrado um trecho de programa para verificar se um determinado número inteiro é um número de Armstrong. Para que o programa funcione como esperado, implemente as seguintes funções:

int expoente (int num); 🡪 recebe como parâmetro um número inteiro e retorna o número de dígitos que compõem esse algarismo.

int Armstrong (int num, int n); 🡪 recebe como parâmetro dois números inteiros. O primeiro, num, representa o número que se deseja verificar se é um número de Armstrong e o segundo, n, significa o número de dígitos que compõem esse número num. A função deve retornar a soma de cada um dos valores de seus dígitos componentes elevado a nésima potência.

int n;

int expoente(int num);

int Armstrong(int num, int n);

main(){

printf("Entre com um número:");

scanf("%d",&n);

if(n==Armstrong(n,expoente(n)))

printf("%d é um número de Armstrong.",n);

else

printf("%d não é um número de Armstrong.",n);

}

int expoente(int num){

int tempo;

tempo=0;

while(num>0){

num/=10;

tempo++;

}

return tempo;

}

int Armstrong(int num, int n){

int Arm,casa;

Arm=0;

for(casa=n;casa>0;casa--){

Arm+=(pow(num%10,n));

num/=10;

}

return Arm;

}

## Passagem de parâmetros

1. Faça o passo a passo e diga o que aparecerá na tela.

a)

int x,y,z;

void imprime1(int \*z, int \*x, int y);

void imprime2(int \*y, int z, int \*x);

main(){

x=1;

y=3;

z=2;

imprime1(&y(3), &z(2), x(1));

printf("1, 6, 9 \n", x, y, z);

imprime2(&x(1), y(6), &z(9));

printf("10, 6, 12 \n", x, y, z);

}

void imprime1(int \*z(3),int \*x(2),int y(1)){

\*z(6)+=\*z(3);

\*x(9)+=y(1)+\*z(6);

y(15)=\*x(9)+\*z(6);

printf("15, 6, 9 \n", y, \*z, \*x);

}

void imprime2(int \*y(1),int z(6),int \*x(9)){

\*y(10)+=\*x(9);

z(10)=\*x(9)+1;

\*x(12)=z(10)+2;

printf("10, 12, 10 \n", z, \*x, \*y);

}

Tela:

15, 6, 9

1, 6, 9

10, 12, 10

10, 6, 12

b)

int x,y,z;

void imprime1(int \*z, int x);

void imprime2(int y, int \*x);

main(){

x=1;

y=3;

z=2;

imprime1(&x(1), z(2));

printf("2, 9, 2 \n", x, y, z);

imprime2(x(2), &y(9));

printf("2, 66, 10 \n", x, y, z);

}

void imprime1(int \*z(1)(11), int x(2)(10)){

\*z(2)(22)+=\*z(1)(11);

x(7)(44)+=y(3)(12)+\*z(2)(22);

y(9)(66)=x(7)(44)+\*z(2)(22);

printf("9(66), 2(22), 7(44) \n", y, \*z, x);

}

void imprime2(int y(2), int \*x(9)){

y(11)+=\*x(9);

z(10)=\*x(9)+1;

\*x(12)=z(10)+2;

printf("10, 12, 11 \n", z, \*x, y);

imprime1(&y(11), z(10));

printf("10, 66, 22 \n", z, \*x, y);

}

9 2 7

2 9 2

10 12 11

66 22 44

10 66 22

2 66 10

int x,y,z;

void imprime2(int y, int \*x);

main(){

x=1;

y=3;

z=2;

imprime2(x(1), &y(3));

printf("\t1\t6\t4\n", x, y, z);

}

void imprime2(int y(1), int \*x(3)){

void imprime1(int \*z(4), int x(4)){

\*z(8)+=\*z(4);

x(20)+=y(8)+\*z(8);

y(28)=x(16)+\*z(8);

printf("\t28\t28\t20\n", y, \*z, x);

}

y(4)+=\*x(3);

z(4)=\*x(3)+1;

\*x(6)=z(4)+2;

printf("\t4\t6\t4\n", z, \*x, y);

imprime1(&y(4), z(4));

printf("\t4\t6\t28\n", z, \*x, y);

} 4 6 4

20 8 16

4 6 8

1 6 4

1. Faça um programa que leia um conjunto de linhas, contendo, cada uma, o número de um empregado, a hora de início (hora, minuto e segundo) e a hora do término (hora, minuto e segundo) de uma determinada tarefa. A última linha conterá o número do empregado igual a zero. Calcule para cada empregado, a duração da tarefa que ele executou. Escreva para cada empregado, o seu número e a duração da sua tarefa em horas, minutos e segundos.

Obs.: Usar uma função do tipo int que receba como parâmetros horas, minutos e segundos e retorne o total de segundos. Usar uma função do tipo void que receba como parâmetro o total de segundos e retorne em horas, minutos e segundos.

int empregado,horaini,horafim,minutoini,minutofim,segundoini,segundofim;

int conversao(int horas, int minutos, int segundos);

void trabalho(int segundos);

main(){

empregado=1;

printf("Para finalizar, basta digitar 0 no numero do empregado.\n");

while(empregado!=0){

printf("Diga o numero do empregado, a hora de inicio e termino da tarefa.\n");

scanf("%d%d%d%d%d%d%d",&empregado,&horaini,&minutoini,&segundoini,&horafim,&minutofim,&segundofim);

horafim-=horaini;

minutofim-=minutoini;

segundofim-=segundoini;

if(empregado==0)

printf("Obrigado por usar meu programa :).");

else

trabalho(conversao(horafim,minutofim,segundofim));

}

}

int conversao(int horas, int minutos, int segundos){

segundos+=horas\*3600;

segundos+=minutos\*60;

return segundos;

}

void trabalho(int segundos){

int horasr,minutosr,segundosr;

segundosr=segundos;

horasr=segundosr/3600;

minutosr=(segundosr%3600)/60;

segundosr=(segundosr%3600)%60;

printf("O empregado %d trabalhou por %d horas, %d minutos e %d segundos.\n\n",empregado,horasr,minutosr,segundosr);

}

## Recursão

1. Faça uma função recursiva que receba um número inteiro e calcule a soma dos n primeiros números inteiros.

int numero;

int soma(int n);

main(){

printf("Diga o numero.\n");

scanf("%d",&numero);

printf("A soma dos %d primeiros numeros e %d.",numero,soma(numero));

}

int soma(int n){

if(n==1)

return 1;

else

return n+soma(n-1);

}

1. Faça uma função recursiva que receba um número inteiro e calcule o n-ésimo termo da série de Fibonacci.

int numero;

int Fibonnachi(int n);

main(){

printf("Diga o numero da sequencia de Fibonnachi.\n");

scanf("%d",&numero);

printf("O %d° numero da sequencia de Fibonnachi e %d.",numero,Fibonnachi(numero));

}

int Fibonnachi(int n){

if(n<=2)

return 1;

else

return Fibonnachi(n-1)+Fibonnachi(n-2);

}

1. Faça uma função recursiva que calcule e retorne o valor de S, onde:

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

int numero;

float S(int n);

main(){

printf("Diga o numero.\n");

scanf("%d",&numero);

printf("O valor de S do numero %d e %.2f.",numero,S(numero));

}

float S(int n){

int fat(int n){

if (n==0)

return 1;

else

return n\*fat(n-1);

}

if(n==0)

return 1;

else

return S(n-1)+1.0/fat(n);

}

1. Faça uma função recursiva que calcule e retorne x elevado a z. Em seguida, apresente o passo a passo de execução do programa com x=2 e z=-2. Obs.: x é um número real e z é um número inteiro. float potencia(float x, int z);

float numero;

int pot;

float potencia(float x, int z);

main(){

printf("Diga o numero e sua potencia.\n");

scanf("%f%d",&numero,&pot);

printf("%.2f elevado a %d e %.2f.",numero,pot,potencia(numero,pot));

}

float potencia(float x, int z){

if(x==0&&z==0)

printf("Erro.");

if(x==0&&z>0)

return 0;

if(x!=0&&z==0)

return 1;

if(x!=0&&z>0)

return x\*potencia(x,z-1);

if(x!=0&&z<0)

return 1/potencia(x,-z);

}

1. Dado o programa abaixo, construa a função recursiva mostra( ), que dado um número n positivo natural, escreva a sequência abaixo. (no exemplo abaixo n=5) 5 4 3 2 1 2 3 4 5

int n;

void mostra(int n);

main(){

printf("Entre com um número: ");

scanf("%d",&n);

mostra(n);

}

void mostra(int n){

if(n==1)

printf("%d ",n);

else{

printf("%d ", n);

mostra(n-1);

printf("%d ",n);

}

}

1. Dado o programa abaixo, construa a função recursiva tab( ), para que apareça na tela a tabuada de 1 até 10 de um número inteiro qualquer.

int n;

void tab(int n, int i);

main(){

printf("Entre com um número: ");

scanf("%d", &n);

tab(n,1);

}

void tab(int n, int i){

if(i==10)

printf("%d\n",n\*i);

else{

printf("%d\n",n\*i);

i++;

tab(n,i);

}

}

1. Os primeiros membros da associação de Pitágoras definiram números poligonais como sendo o número de pontos em determinadas configurações geométricas. Os primeiros números triangulares são 1, 3, 6, 10, 15. Faça uma função recursiva que receba um número natural positivo e encontre o n-ésimo número triangular.

int numero;

int triangular(int num);

main(){

printf("Qual e a posicao do numero triangular?\n");

scanf("%d",&numero);

printf("%d e o numero.",triangular(numero));

}

int triangular(int num){

if(num==1)

return 1;

else

return num+triangular(num-1);

}

1. Os primeiros membros da associação de Pitágoras definiram números poligonais como sendo o número de pontos em determinadas configurações geométricas. Os primeiros números pentagonais são 1, 5, 12, 22. Faça uma função recursiva que receba um número natural positivo e encontre o n-ésimo número pentagonal.

int numero;

int pentagonal(int num);

main(){

printf("Qual e a posicao do numero pentagonal?\n");

scanf("%d",&numero);

printf("%d e o numero.",pentagonal(numero));

}

int pentagonal(int num){

if(num==1)

return 1;

else

return ((num\*3)-2)+pentagonal(num-1);

}

1. Construa uma função recursiva que calcule e retorne o valor de S. S = 1^1! + 2^1! + 3^2! + 4^3! + 5^5! + 6^8! + 7^13! + ... Onde ^ significa elevado. Exemplo 1^1 significa 1 elevado a 1.

int numero;

int S(int rep);

main(){

printf("Qual e a posicao do numero?\n");

scanf("%d",&numero);

printf("%d e o numero na posicao %d.",S(numero),numero);

}

int S(int num){

int Fibonnachi(int n){

if(n<=2)

return 1;

else

return Fibonnachi(n-1)+Fibonnachi(n-2);

}

int fat(int n){

if (n==0)

return 1;

else

return n\*fat(n-1);

}

if(num==1)

return 1;

else

return pow(num,fat(Fibonnachi(num)))+S(num-1);

}

1. Em um experimento, uma determinada colônia de bactérias tem uma população inicial de 50000 bactérias. A população é contada a cada 2 horas e, ao final de cada intervalo de 2 horas, a população triplica. Faça uma função recursiva cont( ), que calcule o número de bactérias presentes no início do n-ésimo período de tempo.

int horas;

int cont(int h);

main(){

printf("Qual e o numero de horas a ser conferido?\n");

scanf("%d",&horas);

printf("Haverao %d bacterias depois de %d horas.",cont(horas),horas);

}

int cont(int h){

if(h<2)

return 50000;

else

return 3\*cont(h-2);

}

1. Uma quantia de R$500,00 é investida em uma aplicação que paga juros de 10% capitalizados anualmente. Faça uma função recursiva P( ), que calcule a quantidade total da aplicação no início do n-ésimo ano. Lembre-se que no início do primeiro ano, tem-se aplicado R$500,00.

int anos;

int P(int tempo);

main(){

printf("Voce quer saber quanto tera em quantos anos?\n");

scanf("%d",&anos);

printf("Voce tera RS%d,00 depois de %d anos.",P(anos),anos);

}

int P(int tempo){

if(tempo==1)

return 500;

else

return P(tempo-1)+(P(tempo-1)\*0.1);

}

1. Complete o que falta para que o programa funcione corretamente. Obs.: Não é permitido criar nenhum tipo de variável.

int n;

int trans(int n, int i);

main(){

printf("Entre com um número decimal: ");

scanf("%d", &n);

printf("%d em binário é igual a %d em decimal.", trans(n,1), n);

}

int trans(int n, int i){

if (n==0)

return 0;

else

return n%2\*i+trans(n/2,i\*10);

}

1. Dado o programa abaixo, reescreva a função primo(), sem utilizar nenhuma estrutura de repetição.

int n;

void primo(int n, int i, int cont);

main(){

printf("Entre com um número: ");

scanf("%d",&n);

primo(n,1,0);

}

void primo(int n, int i, int cont){

if((n%i)==0)

cont++;

if(i==n){

if(cont==2)

printf("Primo.");

else

printf("Nao e primo.");

}

else

primo(n,i+1,cont);

}

1. Dado o programa abaixo, reescreva a função perfeito(), sem utilizar nenhuma estrutura de repetição.

int n;

void perfeito(int n, int i, int cont);

main(){

printf("Entre com um número: ");

scanf("%d",&n);

perfeito(n,1,0);

}

void perfeito(int n, int i, int cont){

if(i==n)

if(cont==n)

printf("E um numero perfeito.");

else

printf("Nao e um numero perfeito.");

else

if((n%i)==0)

cont+=i;

perfeito(n,i+1,cont);

}

1. Complete o que falta para que o programa funcione corretamente. Obs.: Não é permitido criar nenhum tipo de variável.

int n;

int trans(int n, int i);

main(){

printf("Entre com um número binário: ");

scanf("%d", &n);

printf("%d em binário é igual a %d em decimal.", n,trans(n,1));

}

int trans(int n, int i){

if (n==0)

return 0;

else

return (n%10\*i)+trans(n/10,i\*2);

}

## Estrutura de Dados Heterogêneas - Registro

1. Dado os trechos de códigos abaixo, faça as declarações de tipos e variáveis necessárias na linguagem C.

struct disc{

char genero[10];

int quant;

float preco;

};

struct rent{

char endereco[20];

struct disc fita[100];

};

struct rent locadora[100];

int j,k,l,m;

main(){

strcpy(locadora[j].fita[k].genero, "ação");

locadora[l].fita[m].quant = 20;

strcpy(locadora[j].endereco, "Rua B");

locadora[l].fita[m].preco = 15.50;

}

b)

struct pack{

int estoque,peca[20];

};

struct pack produto;

struct depart{

char z[10];

struct pack xxx;

};

struct depart depto;

struct section{

char nome[10];

struct depart indice;

};

struct section setor[20];

int i,k;

main(){

produto.estoque = 2000;

produto.peca[i] = 3;

depto.xxx = produto;

strcpy(depto.z, "informática");

setor[k].indice = depto;

strcpy(setor[k].nome, "almoxarifado");

}

c)

struct client{

char nome[10];

int cpf;

};

struct client cliente[1000];

struct invest{

float entrada,saida;

};

struct ag[100]{

struct invest totinvest;

};

struct ag agencia[50];

struct bank{

char sigla[10];

struct ag agen[100];

};

struct bank banco[50];

int k,i,j;

main(){

strcpy(cliente[k].nome, "Maria");

agencia[i].totinvest.entrada = 30.000,00;

agencia[i].totinvest.saida = 50.000,00;

banco[j].agen[i] = agencia[i];

cliente[k].cpf = 12345;

strcpy(banco[j].sigla, "B145");

}

d)

struct n1{

char nome[13],tipo[13];

int quant;

};

struct n2{

char bairro[10];

struct n1 fitas[100][100];

};

struct n2 locadora[20];

int x,y,i;

main(){

strcpy(locadora[i].fitas[x][y].nome, "X-Men");

strcpy(locadora[i].fitas[x][y].tipo, "ação");

locadora[i].fitas[x][y].quant = 5;

strcpy(locadora[i].bairro, "Tijuca");

}

2) Foi feita uma estatística nas 200 principais cidades brasileiras para coletar dados sobre acidentes de trânsito em 2020. Foram obtidos os seguintes dados:

• Nome da cidade

• Estado (RS, SC, RJ, PR, SP, ...)

• Número de veículos de passeio

• Número de acidentes de trânsito com vítimas

Deseja-se saber:

1. Qual o maior e menor índice de acidentes de trânsito e a que cidades pertencem
2. Qual a média de veículos nas cidades brasileiras
3. Qual a média de acidentes com vítimas nas cidades do Rio de Janeiro.

#include <stdio.h>

#define TAM 3

int maior,menor,tempo,soma,somarj,nrj;

struct cidade{

char nome[20],estado[3];

int veiculos,acidentes;

};

struct cidade cidades[TAM];

main(){

soma=0;

somarj=0;

nrj=0;

for(tempo=0;tempo<TAM;tempo++){

printf("\n\nDiga o nome e a sigla do estado da proxima cidade.\n");

fflush(stdin);

gets(cidades[tempo].nome);

fflush(stdin);

gets(cidades[tempo].estado);

printf("Diga agora o numero de veiculos e de acidentes de transito.\n");

scanf("%d",&cidades[tempo].veiculos);

scanf("%d",&cidades[tempo].acidentes);

soma+=cidades[tempo].veiculos;

if(strcmp("RJ",cidades[tempo].estado)==0){

somarj+=cidades[tempo].acidentes;

nrj++;

}

if(tempo==0){

maior=cidades[tempo].acidentes;

menor=cidades[tempo].acidentes;

}

else{

if(cidades[tempo].acidentes>maior)

maior=cidades[tempo].acidentes;

if(cidades[tempo].acidentes<menor)

menor=cidades[tempo].acidentes;

}

}

for(tempo=0;tempo<TAM;tempo++){

if(maior==cidades[tempo].acidentes)

printf("%s foi a cidade com mais acidentes, tendo % acidentes.\n",cidades[tempo].nome,cidades[tempo].acidentes);

if(menor==cidades[tempo].acidentes)

printf("%s foi a cidade com menos acidentes, tendo apenas %d acidentes.\n",cidades[tempo].nome,maior);

}

printf("\nA media de veiculos nas cidades analisadas e %d veiculos por cidade.",soma/TAM);

printf("\nA media de acidentes nas cidades do RJ e %d acidentes por cidade.",somarj/nrj);

}

1. Um escritório de advocacia, especializada em direito de família, deseja fazer um sistema para automatizar todo o seu processo. Você foi designado para realizar tal tarefa. Faça um programa que receba as seguintes informações de 1000 clientes:

 nome (100 posições)

 sexo (feminino-1 e masculino-2)

 estado civil (solteiro =1, casado=2, divorciado=3, viúvo=4)

 idade

 salário do pai

 salário do cônjuge (se não for casado o salário é zero)

Com base no vetor de registro criado, faça um programa que crie um segundo vetor de registro

contendo todas essa informações, mais a pensão a que tem direito, sabendo-se que:

 caso seja homem, solteiro e tenha idade inferior a 18 anos, tem direito a pensão do pai

(15% do salário do pai)

 caso seja mulher (sexo=1) e tenha idade inferior a 21 anos tem direito a pensão do pai

(20% do salário do pai)

 se for solteira; caso contrário terá direito a pensão do cônjuge (30% do salário do

cônjuge) se for casada.

 Se não atender a nenhuma das condições acima, a pessoa não terá direito a pensão

(pensão será zero)

Ao final mostre o conteúdo do segundo vetor na tela em formato de tabela, mostrando sexo e

estado civil por extenso.

#include <stdio.h>

#define TAM 1000

struct ind{

char nome[100];

int sexo,estciv,idade;

float salpai,salconj;

};

struct dir{

struct ind pessoa[TAM];

float pensao;

};

struct dir cliente[TAM];

int tempo;

main(){

for(tempo=0;tempo<TAM;tempo++){

printf("\nInsira seu nome.\n");

fflush(stdin);

gets(cliente[tempo].pessoa[tempo].nome);

printf("\nAgora, insira seu sexo, 1 para feminino e 2 para masculino.\n");

scanf("%d",&cliente[tempo].pessoa[tempo].sexo);

while(cliente[tempo].pessoa[tempo].sexo>2||cliente[tempo].pessoa[tempo].sexo<1){

printf("\nInsira uma resposta valida.\n");

scanf("%d",&cliente[tempo].pessoa[tempo].sexo);

}

printf("\nInsira seu estado civil, 1 para solteiro, 2 para casado, 3 para divorciado e 4 para viuvo.\n");

scanf("%d",&cliente[tempo].pessoa[tempo].estciv);

while(cliente[tempo].pessoa[tempo].estciv>4||cliente[tempo].pessoa[tempo].estciv<1){

printf("\nInsira uma resposta valida.\n");

scanf("%d",&cliente[tempo].pessoa[tempo].estciv);

}

printf("\nInsira sua idade.\n");

scanf("%d",&cliente[tempo].pessoa[tempo].idade);

if(cliente[tempo].pessoa[tempo].estciv==2){

printf("\nPor fim, insira o salario do seu pai e do seu conjuge.\n");

scanf("%f%f",&cliente[tempo].pessoa[tempo].salpai,&cliente[tempo].pessoa[tempo].salconj);

}

else{

printf("\nPor fim, insira o salario do seu pai.\n");

scanf("%f",&cliente[tempo].pessoa[tempo].salpai);

cliente[tempo].pessoa[tempo].salconj=0;

}

if(cliente[tempo].pessoa[tempo].sexo==2)

if(cliente[tempo].pessoa[tempo].estciv==1&&cliente[tempo].pessoa[tempo].idade<18)

cliente[tempo].pensao=0.15\*cliente[tempo].pessoa[tempo].salpai;

else

cliente[tempo].pensao=0;

else

if(cliente[tempo].pessoa[tempo].estciv==2)

cliente[tempo].pensao=0.3\*cliente[tempo].pessoa[tempo].salconj;

else

if(cliente[tempo].pessoa[tempo].estciv==1&&cliente[tempo].pessoa[tempo].idade<21)

cliente[tempo].pensao=0.2\*cliente[tempo].pessoa[tempo].salpai;

else

cliente[tempo].pensao=0;

}

printf("\n\n\nNome:\tIdade:\tSexo:\tEstado civil\tSalario do pai:\tSalario do conjuge:\tPensao:\n");

for(tempo=0;tempo<TAM;tempo++){

printf("%s\t%d\t",cliente[tempo].pessoa[tempo].nome,cliente[tempo].pessoa[tempo].idade);

if(cliente[tempo].pessoa[tempo].sexo==1)

printf("Mulher\t");

else

printf("Homem\t");

if(cliente[tempo].pessoa[tempo].estciv==1)

printf("Solteiro\t");

if(cliente[tempo].pessoa[tempo].estciv==2)

printf("Casado\t");

if(cliente[tempo].pessoa[tempo].estciv==3)

printf("Divorciado\t");

if(cliente[tempo].pessoa[tempo].estciv==4)

printf("Viuvo\t");

printf("%.2f\t%.2f\t%.2f\n",cliente[tempo].pessoa[tempo].salpai,cliente[tempo].pessoa[tempo].salconj,cliente[tempo].pensao);

}

}

1. Faça um programa que gerencie o estoque de um mercado e:

* crie e leia um vetor de 5 produtos, com os dados: código (inteiro), nome (máximo 15 letras), preço e quantidade.
* leia um pedido, composto por um código de produto e a quantidade. Localize este código no vetor e, se houver quantidade suficiente para atender ao pedido integralmente, atualize o estoque, e informe o usuário o valor total do pedido realizado.

Repita este processo até ler um código igual a zero.

Se por algum motivo não for possível atender ao pedido, mostre uma mensagem informando qual erro ocorreu.

#include <stdio.h>

#define TAM 5

struct produto{

int code,quantidade;

float preco;

char nome[16];

};

struct produto prod[TAM];

int tempo,codigo,quant;

main(){

for(tempo=0;tempo<TAM;tempo++){

printf("\nInsira o codigo do produto.\n");

scanf("%d",&prod[tempo].code);

printf("Insira o nome do produto.\n");

fflush(stdin);

gets(prod[tempo].nome);

printf("Insira o preco do produto.\n");

scanf("%f",&prod[tempo].preco);

printf("Insira a quantidade do produto.\n");

scanf("%d",&prod[tempo].quantidade);

}

do{

printf("\nInsira o codigo e a quantidade do produto desejado.\n");

scanf("%d%d",&codigo,&quant);

for(tempo=0;tempo<TAM;tempo++)

if(codigo==prod[tempo].code)

if(quant<=prod[tempo].quantidade){

prod[tempo].quantidade=prod[tempo].quantidade-quant;

printf("O valor a ser pago e de R$%.2f.\n",prod[tempo].preco\*quant);

}

else

printf("Quantidade indisponivel.\n");

else

if(tempo==TAM-1)

printf("Nao ha nenhum produto com esse codigo no estoque.\n");

}while(codigo!=0);

}

## Estruturas de dados dinâmicas - Ponteiro

1. Faça um programa que leia um número inteiro e diga se é par ou ímpar.

#include<stdio.h>

int \*ponteiro;

main(){

ponteiro=(int\*)malloc(sizeof(int));

printf("Diga um numero inteiro.\n");

scanf("%d",&\*ponteiro);

if(\*ponteiro%2)

printf("%d e impar.",\*ponteiro);

else

printf("%d e par.",\*ponteiro);

free(ponteiro);

}

1. Faça um programa que leia um número inteiro e diga se é um número primo.

#include<stdio.h>

#include<stdlib.h>

int \*n,\*tempo,\*cont;

main(){

n=(int \*)malloc(sizeof(int));

tempo=(int \*)malloc(sizeof(int));

cont=(int \*)malloc(sizeof(int));

printf("Insira um numero.\n");

scanf("%d",n);

\*cont=0;

for((\*tempo)=1;(\*tempo)<=(\*n);(\*tempo)++)

if((\*n)%(\*tempo)==0)

(\*cont)++;

if((\*cont)>2)

printf("%d nao e primo.",\*n);

else

printf("%d e primo.",\*n);

free(n);

free(tempo);

free(cont);

}

1. Faça um programa que leia um número inteiro e calcule o seu fatorial.

#include<stdio.h>

#include<stdlib.h>

int \*n,\*tempo;

main(){

n=(int \*)malloc(sizeof(int));

tempo=(int \*)malloc(sizeof(int));

printf("Diga o numero inteiro.\n");

scanf("%d",n);

if((\*n)==0)

\*n=1;

if((\*n)<0)

printf("Esse numero e negativo.");

else{

for((\*tempo)=\*n;(\*tempo)>=2;(\*tempo)--)

(\*n)=(\*n)\*((\*tempo)-1);

printf("O fatorial do numero e %d.",\*n);

}

free(n);

free(tempo);

}

1. Faça um programa que leia um número inteiro e calcule e mostre os n primeiros termos da série de Fibonacci.

#include<stdio.h>

#include<stdlib.h>

int \*n,\*tempo,\*resposta,\*n1,\*n2;

main(){

n=(int \*)malloc(sizeof(int));

tempo=(int \*)malloc(sizeof(int));

resposta=(int \*)malloc(sizeof(int));

n1=(int \*)malloc(sizeof(int));

n2=(int \*)malloc(sizeof(int));

printf("Quantos numeros da sequencia fibonacci serao apresentados?\n");

scanf("%d",n);

\*n1=1;

\*n2=0;

\*resposta=1;

for((\*tempo)=0;(\*tempo)<=(\*n);(\*tempo)++){

printf("%d, ",\*resposta);

(\*resposta)=(\*n1)+(\*n2);

(\*n2)=(\*n1);

(\*n1)=(\*resposta);

}

free(n);

free(tempo);

free(resposta);

free(n1);

free(n2);

}

1. Faça um programa que leia um número binário qualquer e calcule o seu decimal correspondente.

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int \*binario, \*decimal, \*ncasa, \*casa;

main(){

binario=(int \*)malloc(sizeof(int));

decimal=(int \*)malloc(sizeof(int));

ncasa=(int \*)malloc(sizeof(int));

casa=(int \*)malloc(sizeof(int));

printf("Diga o numero binario.\n");

scanf("%d",binario);

if((\*binario)==0)

printf("Esse numero em decimal e %d.",\*binario);

else{

\*decimal=\*binario;

\*ncasa=0;

while((\*decimal)!=0){

(\*ncasa)++;

(\*decimal)/=10;

}

\*decimal=0;

\*casa=0;

while((\*casa)<=(\*ncasa)){

if((\*binario)%10)

\*decimal=(\*decimal)+pow(2,\*casa);

(\*binario)/=10;

(\*casa)++;

}

printf("Esse numero em decimal e %d.",\*decimal);

}

free(binario);

free(decimal);

free(ncasa);

free(casa);

}

1. Faça um programa que leia a matrícula (inteiro) e nota (real) de 5 alunos e em seguida mostre na tela as matrículas dos alunos que obtiveram a nota maior ou igual a média dos 5 alunos. Obs.: Não utilize vetor e crie uma estrutura contendo a matrícula e a nota para cada aluno.

#include<stdio.h>

#include<stdlib.h>

struct matricula{

int matricula;

float nota;

};

struct matricula \*aluno1,\*aluno2,\*aluno3,\*aluno4,\*aluno5;

float \*media;

main(){

aluno1=(struct matricula \*)malloc(sizeof(struct matricula));

aluno2=(struct matricula \*)malloc(sizeof(struct matricula));

aluno3=(struct matricula \*)malloc(sizeof(struct matricula));

aluno4=(struct matricula \*)malloc(sizeof(struct matricula));

aluno5=(struct matricula \*)malloc(sizeof(struct matricula));

media=(float \*)malloc(sizeof(float));

printf("Entre com a matricula e a nota do aluno 1.\n");

scanf("%d%f",&aluno1->matricula,&aluno1->nota);

printf("Entre com a matricula e a nota do aluno 2.\n");

scanf("%d%f",&aluno2->matricula,&aluno2->nota);

printf("Entre com a matricula e a nota do aluno 3.\n");

scanf("%d%f",&aluno3->matricula,&aluno3->nota);

printf("Entre com a matricula e a nota do aluno 4.\n");

scanf("%d%f",&aluno4->matricula,&aluno4->nota);

printf("Entre com a matricula e a nota do aluno 5.\n");

scanf("%d%f",&aluno5->matricula,&aluno5->nota);

\*media=((aluno1->nota)+(aluno2->nota)+(aluno3->nota)+(aluno4->nota)+(aluno5->nota))/5;

if(aluno1->nota>\*media)

printf("%d\n",aluno1->matricula);

if(aluno2->nota>\*media)

printf("%d\n",aluno2->matricula);

if(aluno3->nota>\*media)

printf("%d\n",aluno3->matricula);

if(aluno4->nota>\*media)

printf("%d\n",aluno4->matricula);

if(aluno5->nota>\*media)

printf("%d\n",aluno5->matricula);

free(media);

free(aluno1);

free(aluno2);

free(aluno3);

free(aluno4);

free(aluno5);

}

1. Faça um programa que leia a altura e o sexo (codificado em 1=masculino e 2=feminino) de um grupo de n pessoas lidas do teclado. Calcule e escreva:

a) Média da altura dos membros do grupo

b) Média da altura dos homens

c) Média da altura das mulheres

d) Maior altura do grupo

e) Menor altura do grupo

Obs.: crie uma estrutura contendo a altura e o sexo de cada uma das n pessoas.

#include<stdio.h>

#include<stdlib.h>

int \*tempo,\*n,\*sexo,\*m,\*f;

float \*altura,\*mediag,\*mediaf,\*mediam,\*maior,\*menor;

main(){

tempo=(int \*)malloc(sizeof(int));

n=(int \*)malloc(sizeof(int));

sexo=(int \*)malloc(sizeof(int));

m=(int \*)malloc(sizeof(int));

f=(int \*)malloc(sizeof(int));

altura=(float \*)malloc(sizeof(float));

mediag=(float \*)malloc(sizeof(float));

mediaf=(float \*)malloc(sizeof(float));

mediam=(float \*)malloc(sizeof(float));

maior=(float \*)malloc(sizeof(float));

menor=(float \*)malloc(sizeof(float));

\*m=0;

\*f=0;

printf("Diga quantas pessoas fazem parte do grupo.\n");

scanf("%d",n);

for(\*tempo=1;\*tempo<=\*n;(\*tempo)++){

printf("Diga o sexo e a altura, masculino e 1 e feminino e 2.\n");

scanf("%d%f",sexo,altura);

\*mediag+=\*altura;

if(\*sexo==1){

(\*mediam)+=\*altura;

(\*m)++;

}

if(\*sexo==2){

(\*mediaf)+=\*altura;

(\*f)++;

}

if(\*tempo==1){

\*menor=\*altura;

\*maior=\*altura;

}

if(\*altura>\*maior)

\*maior=\*altura;

if(\*altura<\*menor)

\*menor=\*altura;

}

if(\*m==0)

\*m=1;

if(\*f==0)

\*f=1;

printf("\nA media da altura do grupo e: %.2f,\nA media da altura das mulheres e %.2f.\nA media da altura dos homens e %.2f.\nA maior altura e %.2f e a menor e %.2f.",(\*mediag/ \*n),(\*mediaf/ \*f),(\*mediam/ \*m),\*maior,\*menor);

free(tempo);

free(n);

free(sexo);

free(altura);

free(m);

free(f);

free(mediag);

free(mediaf);

free(mediam);

free(maior);

free(menor);

}

## Listas simplesmente encadeadas

#include<stdio.h>

#include<stdlib.h>

struct list{

int value;

struct list \*next;

};

struct list \*inicio;

void ldelete(struct list \*aux);

void lnew(int tempo);

void lshow(int tempo,struct list \*aux);

int lnumber\_elements(int tempo,struct list \*aux);

void lbiggest\_smallest(int tempo,struct list \*aux,int biggest,int smallest);

float lmedia(int tempo,struct list \*aux, float media);

void leinsert(int tempo,struct list \*aux,int nvalue);

void lsinsert(int tempo,struct list \*aux,struct list \*smallest,struct list \*biggest,int nvalue);

main(){

lnew(0);

lshow(0,inicio);

printf("\nA lista possui %d elementos.",lnumber\_elements(0,inicio));

lbiggest\_smallest(0,inicio,0,0);

lmedia(0,inicio,0);

printf("\nA media dos elementos da lista e %.2f.", lmedia(0,inicio,0));

leinsert(0,inicio,0);

lshow(0,inicio);

lsinsert(0,inicio,0,inicio,0);

lshow(0,inicio);

ldelete(inicio);

}

void ldelete(struct list \*aux){

if(inicio!=(struct list \*)NULL){

inicio=inicio->next;

aux->next=(struct list \*)NULL;

free(aux);

ldelete(inicio);

}

}

//1. Geração de uma LSE onde os elementos são inseridos na ordem inversa em que são obtidos. Condição de parada é um valor negativo.

void lnew(int tempo){

struct list \*aux;

int nvalue;

if(tempo=0){

inicio=(struct list \*)NULL;

}

printf("Qual e o valor do proximo elemento da lista?\n");

scanf("%d",&nvalue);

if(nvalue>=0){

aux=(struct list \*)malloc(sizeof(struct list));

aux->value=nvalue;

aux->next=inicio;

inicio=aux;

aux=(struct list \*)NULL;

lnew(tempo+1);

}

}

//2. Mostrar o conteúdo de uma LSE na tela.

void lshow(int tempo,struct list \*aux){

if((tempo==0)&&(aux==(struct list \*)NULL))

printf("\nLista vazia.");

if(aux!=(struct list \*)NULL){

printf("%d\t",aux->value);

lshow(tempo+1,aux->next);

}

}

//3. Conta o número de elementos da LSE.

int lnumber\_elements(int tempo,struct list \*aux){

if(aux!=(struct list \*)NULL)

lnumber\_elements(tempo+1,aux->next);

else

return tempo;

}

//4. Maior e Menor Elemento de uma LSE.

void lbiggest\_smallest(int tempo,struct list \*aux,int biggest,int smallest){

if(tempo==0)

if(aux==(struct list \*)NULL){

tempo=-1;

printf("\nLista vazia, nao ha maior ou menor elemento.");

}

else{

biggest=aux->value;

smallest=aux->value;

}

if(tempo!=-1)

if(aux!=(struct list \*)NULL){

if(biggest<aux->value)

biggest=aux->value;

if(smallest>aux->value)

smallest=aux->value;

lbiggest\_smallest(tempo+1,aux->next,biggest,smallest);

}

else

printf("\n%d e o maior numero da lista e %d e o menor.",biggest,smallest);

}

//5. Média dos Elementos de uma LSE.

float lmedia(int tempo,struct list \*aux, float media){

if((tempo==0)&&(aux==(struct list \*)NULL))

return media;

else

if(aux!=(struct list \*)NULL)

lmedia(tempo+1,aux->next,media+aux->value);

else

return media/tempo;

}

//6. Inserção de um elemento no fim de uma LSE

void leinsert(int tempo,struct list \*aux,int nvalue){

if(tempo==0){

printf("\nQual e o valor que sera inserido no final da lista?\n");

scanf("%d",&nvalue);

}

if(inicio==(struct list \*)NULL){

inicio=(struct list \*)malloc(sizeof(struct list));

inicio->value=nvalue;

inicio->next=(struct list \*)NULL;

}

else

if(aux->next!=(struct list \*)NULL)

leinsert(tempo+1,aux->next,nvalue);

else{

aux->next=(struct list \*)malloc(sizeof(struct list));

aux=aux->next;

aux->value=nvalue;

aux->next=(struct list \*)NULL;

aux=(struct list \*)NULL;

}

}

//7. Inserção de um elemento na LSE mantendo a LSE ordenada. Supor que se a LSE existir, ela está ordenada.

void lsinsert(int tempo,struct list \*aux,struct list \*smallest,struct list \*biggest,int nvalue){

if(tempo==0){

printf("\nQual e o valor do elemento a ser inserido na lista?\n");

scanf("%d",&nvalue);

aux=(struct list \*)malloc(sizeof(struct list));

aux->value=nvalue;

aux->next=(struct list \*)NULL;

}

if(inicio==(struct list \*)NULL){

inicio=aux;

smallest=(struct list \*)NULL;

biggest=(struct list \*)NULL;

aux=(struct list \*)NULL;

}

else

if(inicio->value>nvalue){

aux->next=inicio;

inicio=aux;

aux=(struct list \*)NULL;

}

else

if((biggest->value<nvalue)&&(biggest->next!=(struct list \*)NULL))

lsinsert(tempo+1,aux,biggest,biggest->next,nvalue);

else

if(nvalue>biggest->value){

biggest->next=aux;

smallest=(struct list \*)NULL;

biggest=(struct list \*)NULL;

aux=(struct list \*)NULL;

}

else{

smallest->next=aux;

aux->next=biggest;

smallest=(struct list \*)NULL;

biggest=(struct list \*)NULL;

aux=(struct list \*)NULL;

}

}

#include<stdio.h>

#include<stdlib.h>

struct list{

int value;

struct list \*next;

};

struct list \*inicio;

void lshow(int tempo,struct list \*aux);

void lnewo(int tempo);

void leledelete(int tempo,struct list \*smallest,struct list \*biggest,int nvalue);

void ldelete(struct list \*aux);

void lnews(int tempo,struct list \*aux);

main(){

lnewo(0);

lshow(0,inicio);

leledelete(0,inicio,inicio,0);

lshow(0,inicio);

ldelete(inicio);

lnews(0,0);

lshow(0,inicio);

ldelete(inicio);

}

void lshow(int tempo,struct list \*aux){

if((tempo==0)&&(aux==(struct list \*)NULL))

printf("Lista vazia.");

if(aux!=(struct list \*)NULL){

printf("%d\t",aux->value);

lshow(tempo+1,aux->next);

}

}

//8. Criação de uma LSE em ordem crescente de valor.

void lnewo(int tempo){

void lsinsert(int tempo,struct list \*aux,struct list \*smallest,struct list \*biggest,int nvalue){

if(tempo==0){

aux=(struct list \*)malloc(sizeof(struct list));

aux->value=nvalue;

aux->next=(struct list \*)NULL;

}

if(inicio==(struct list \*)NULL){

inicio=aux;

smallest=(struct list \*)NULL;

biggest=(struct list \*)NULL;

aux=(struct list \*)NULL;

}

else

if(inicio->value>nvalue){

aux->next=inicio;

inicio=aux;

aux=(struct list \*)NULL;

}

else

if((biggest->value<nvalue)&&(biggest->next!=(struct list \*)NULL))

lsinsert(tempo+1,aux,biggest,biggest->next,nvalue);

else

if(nvalue>biggest->value){

biggest->next=aux;

smallest=(struct list \*)NULL;

biggest=(struct list \*)NULL;

aux=(struct list \*)NULL;

}

else{

smallest->next=aux;

aux->next=biggest;

smallest=(struct list \*)NULL;

biggest=(struct list \*)NULL;

aux=(struct list \*)NULL;

}

}

int nvalue;

if(tempo=0)

inicio=(struct list \*)NULL;

printf("\nInsira o valor de um elemento da lista. Insira um negativo para parar.\n");

scanf("%d",&nvalue);

if(nvalue>=0){

lsinsert(0,inicio,0,inicio,nvalue);

lnewo(tempo++);

}

}

//9. Remoção de um elemento de uma LSE.

void leledelete(int tempo,struct list \*smallest,struct list \*biggest,int nvalue){

if(inicio==(struct list \*)NULL)

printf("\nLista vazia, nao ha elemento para ser removido.");

else

if(tempo==0){

printf("\nQual e o valor do elemento a ser retirado da lista?\n");

scanf("%d",&nvalue);

}

if(inicio->next==(struct list \*)NULL){

if(nvalue==inicio->value){

free(inicio);

inicio=(struct list \*)NULL;

}

else

printf("Nao ha um elemento com esse valor na lista.");

}

else{

if(nvalue==inicio->value){

inicio=inicio->next;

smallest->next=(struct list \*)NULL;

free(smallest);

}

else{

if((nvalue!=biggest->value)&&(biggest->next!=(struct list \*)NULL))

leledelete(tempo+1,biggest,biggest->next,nvalue);

else

if(nvalue==biggest->value){

smallest->next=biggest->next;

biggest->next=(struct list \*)NULL;

free(biggest);

}

else

printf("Nao ha um elemento com esse valor na lista.");

}

}

biggest=(struct list \*)NULL;

smallest=(struct list \*)NULL;

}

//10. Remoção de toda LSE da memória.

void ldelete(struct list \*aux){

if(inicio!=(struct list \*)NULL){

inicio=inicio->next;

aux->next=(struct list \*)NULL;

free(aux);

ldelete(inicio);

}

}

//11. Criação de uma LSE na ordem em que os elementos foram inseridos pelo usuário.

void lnews(int tempo,struct list \*aux){

int nvalue;

printf("\nInsira o proximo valor da lista. Insira um numero negativo para parar.\n");

scanf("%d",&nvalue);

if(tempo==0){

inicio=(struct list \*)NULL;

if(nvalue>=0){

inicio=(struct list \*)malloc(sizeof(struct list));

inicio->value=nvalue;

inicio->next=(struct list \*)NULL;

lnews(tempo+1,inicio);

}

}

else

if(nvalue>=0){

aux->next=(struct list \*)malloc(sizeof(struct list));

aux=aux->next;

aux->value=nvalue;

aux->next=(struct list \*)NULL;

lnews(tempo+1,aux);

}

else

aux=(struct list \*)NULL;

}