

Objetivos da disciplina

- Desenvolver a lógica de programação através de uma linguagem de programação estruturada.
- Escolher a melhor estrutura e o melhor algoritmo para a solução de um determinado problema.
- Conceituar variável, estrutura de decisão, estrutura de repetição no contexto de algoritmos.
- Avaliar algoritmos utilizando teste de mesa.
- Implementar algoritmos simples utilizando uma linguagem de programação.

Bibliografias

- ASCENCIO, Ana Fernanda Gomes. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ (padrão ANSI) e Java. & CAMPOS, Edilene Aparecida Veneruchi de. 3ª Edição. São Paulo: Person Education do Brasil, 2012.
- MIZRAHI, Victorine Viviane. Treinamento em Linguagem C. 2ª edição. São Paulo: Pearson Prentice Hall, 2008.
- GUEDES, Sergio. Lógica de programação algorítmica. São Paulo: Pearson Education do Brasil, 2014.

O que é lógica?

O que é um algoritmo?



“Como você escovou os dentes hoje?”

“Como fazer um bolo de chocolate?”

“Como trocar uma lâmpada?”



SÍMBOLOS UTILIZADOS NO FLUXOGRAMA



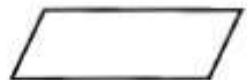
Símbolo utilizado para indicar o início e o fim do algoritmo.



Permite indicar o sentido do fluxo de dados. Serve exclusivamente para conectar os símbolos ou blocos existentes.



Símbolo utilizado para indicar cálculos e atribuições de valores.



Símbolo utilizado para representar a entrada de dados.



Símbolo utilizado para representar a saída de dados.

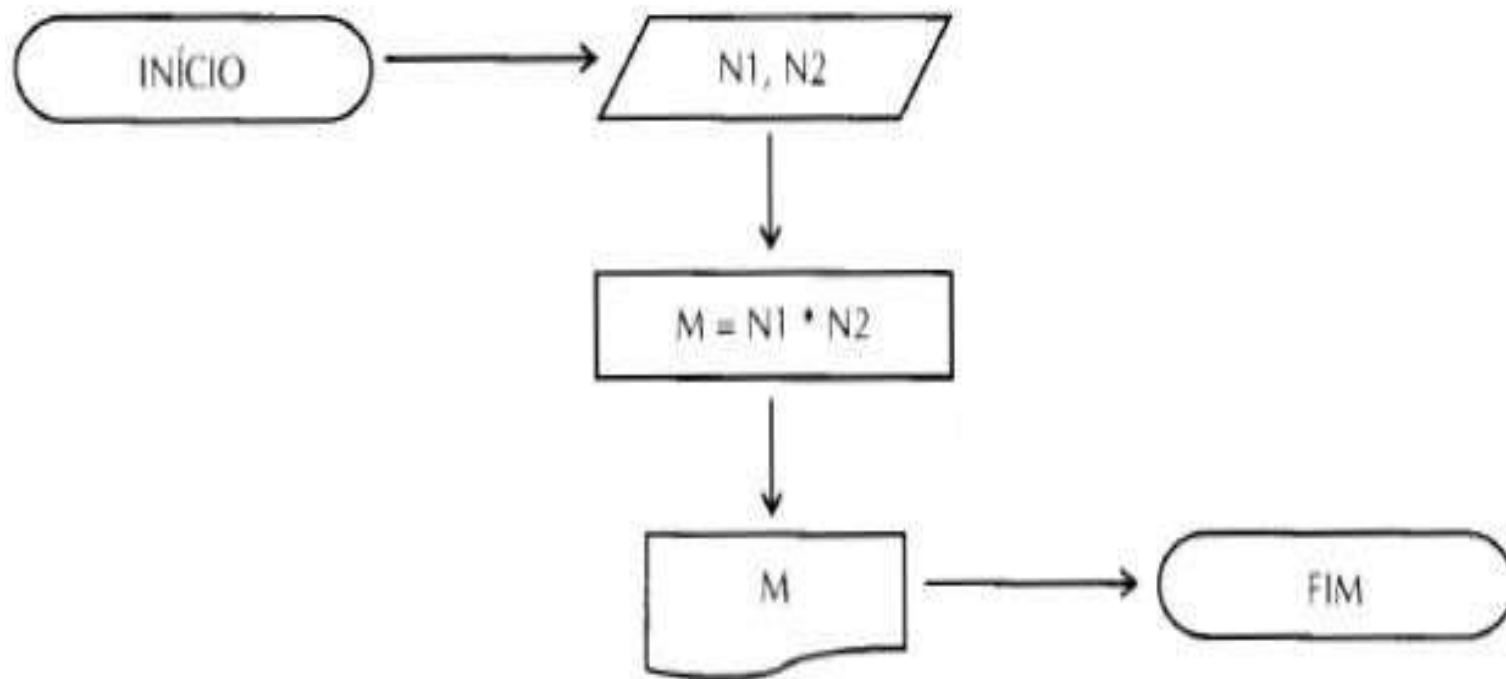


Símbolo que indica que deve ser tomada uma decisão, indicando a possibilidade de desvios.

EXERCÍCIO

Faça um algoritmo para mostrar o resultado da multiplicação de dois números.

RESOLUÇÕES



EXERCÍCIO

Faça um algoritmo para mostrar o resultado da média aritmética entre dois números.

Imprimir a frase: Uma frase

```
#include<stdio.h>

int main(){

printf("Olá mundo\n");

return 0;
}
```

Imprimir a frase: Duas frases

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

```
int main(){
```

```
printf("Hello world\n");
```

```
printf("Olá mundo\n");
```

```
return 0;
```

```
}
```

CONCEITOS IMPORTANTES

// significa que é um comentário, pode ser uma frase, ou até mesmo uma linha de comando que não se deseja executar.

Variáveis: um espaço destinado ao armazenamento de um determinado tipo de dado.

Tipos de dados:

- **int:** variáveis numéricas do tipo inteiro, ou seja, sem casas decimais
- **float ou double:** variáveis numéricas do tipo real, ou seja, com casas decimais.
- **char:** variável do tipo string, ou seja, cadeia de caracteres.

Pseudocódigos -> contém os comandos relativos à execução do algoritmo.

Declaração de variável:

tipo da variável nome da variável;

int idade;

float altura, peso;

Formação de Identificadores: Nome de variável

- **NOMES VÁLIDOS:** Alpha, Media, X2345, Nome_Aluno, A_10, Curso, Salario.
- **NOMES NÃO VÁLIDOS:** X-Y, Nota/2, PLRTX**, Nome Aluno, A*B, Salário, @luno.

COMANDOS DE ENTRADA E SAÍDA

Atribuição automática:

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

```
int a;
```

```
int main(){
```

```
a=6;
```

```
printf("O valor de a = %d\n",a);
```

```
return 0;
```

```
}
```

COMANDOS DE ENTRADA E SAÍDA

Atribuição feita pelo usuário:

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

```
int a;
```

```
int main(){
```

```
printf("Digite o valor de a\n");
```

```
scanf("%d",&a);
```

```
printf("O valor de a = %d\n",a);
```

```
return 0;
```

```
}
```

COMANDOS DE ENTRADA E SAÍDA

Faça um algoritmo que leia o nome, a idade, o peso e o sexo de uma pessoa. Imprimir todas as informações no final.

Dicas:

use %f para variável float

use %lf para variável double

use %d para variável int

use %c para variável char (um único caractere)

use %s para variável char (cadeia de caracteres)

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

```
int idade;
```

```
float peso;
```

```
char sexo, nome[40];
```

```
int main(){
```

```
printf("Digite sua idade\n");
```

```
scanf("%d",&idade);
```

```
printf("\nDigite seu peso\n");
```

```
scanf("%f",&peso);
```

```
printf("\nDigite seu sexo\n");
```

```
scanf(" %c",&sexo);
```

```
printf("\nDigite seu nome\n");
```

```
scanf(" %[^\n]",&nome);
```

```
printf("\nIdade: %d, Peso: %2.2f, Sexo: %c, Nome: %s\n",idade,peso,sexo,nome);
```

```
return 0;
```

```
}
```

Resolução

OPERAÇÃO ARITMÉTICAS

Faça um algoritmo que receba dois números real e calcule e imprima a soma, subtração, multiplicação e divisão.

Operações aritméticas: +, -, *, /

Resolução

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

```
float a,b;
```

```
int main(){
```

```
printf("Digite o valor 1: \n");
```

```
scanf("%f",&a);
```

```
printf("Digite o valor 2: \n");
```

```
scanf("%f",&b);
```

```
printf("\n%.2f\n",a/b);
```

```
printf("\n%.2f\n",a*b);
```

```
printf("\n%.2f\n",a+b);
```

```
printf("\n%.2f\n",a-b);
```

```
return 0;
```

```
}
```

OPERAÇÃO ARITMÉTICAS

Faça um algoritmo que receba a nota 1 e a nota 2 do aluno e calcule e imprima a média aritmética do aluno.

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

```
float n1,n2,med;
```

```
int main(){
```

```
printf("Digite a nota 1: \n");
```

```
scanf("%f",&n1);
```

```
printf("Digite a nota 2: \n");
```

```
scanf("%f",&n2);
```

```
med=(n1+n2)/2;
```

```
printf("\nA média é: %.2f\n",med);
```

```
return 0;
```

```
}
```

Resolução

OPERAÇÃO ARITMÉTICAS

Construir um algoritmo que calcule a área de um triângulo dado a base e a altura.

Fórmula:

$$\text{area} = (\text{base} * \text{altura}) / 2.$$

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

```
float area,base,altura;
```

```
int main(){
```

Resolução

```
printf("Digite o tamanho da base:\n");
```

```
scanf("%f",&base);
```

```
printf("Digite o tamanho da altura:\n");
```

```
scanf("%f",&altura);
```

```
area=(base*altura)/2;
```

```
printf("\nA área do triângulo é: %.2f\n",area);
```

```
return 0;
```

```
}
```

OPERAÇÃO ARITMÉTICAS

Construir um algoritmo que leia o ano de nascimento de uma pessoa e o ano atual, calcule e mostre:

- A idade dessa pessoa;
- Quantos anos essa pessoa terá em 2030.

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

```
int ano_nasc, ano_atual, idade, idadef;
```

```
int main(){
```

```
printf("Digite o ano que você nasceu\n");
```

```
scanf("%d",&ano_nasc);
```

```
printf("\nDigite o ano atual\n");
```

```
scanf("%d",&ano_atual);
```

```
idade = ano_atual - ano_nasc;
```

```
idadef = 2030-ano_nasc;
```

```
printf("\nA idade atual é: %d e em 2030 será %d\n",idade,idadef);
```

```
return 0;
```

```
}
```

Resolução

OPERAÇÃO ARITMÉTICAS

Faça um programa que receba o salário de um funcionário, calcule e mostre o novo salário, sabendo-se que este sofreu um aumento de 25%.

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

```
float sal,novosal;
```

```
int main(){
```

```
printf("Digite seu salário\n");
```

```
scanf("%f",&sal);
```

```
novosal=sal+sal*0.25; //forma 1
```

```
printf("\nO novo salário é: %.2f\n",novosal);
```

```
return 0;
```

```
}
```

```
novosal=sal+((sal*25)/100); //forma 1
```

```
novosal=sal *1.25; // forma 2
```

```
aumento=sal * 0.25 ; // forma 3
```

```
novoval=sal+aumento;
```

Resolução

OPERAÇÃO ARITMÉTICAS

Na loja X a TV está com 10% de desconto. Faça um programa que receba o valor da televisão sem o desconto, calcule e mostre o valor da TV com o desconto.

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

```
float valor,novoval;
```

```
int main(){  
printf("Digite o valor da TV:\n");  
scanf("%f",&valor);
```

Resolução

```
novoval = valor - valor * 0.10; // forma 1
```

```
printf("\nO valor da TV com desconto é: %.2f\n",novoval);  
return 0;  
}
```

```
novoval = valor - ((valor * 10)/100); // forma 2
```

```
novoval=valor *0.90; // forma 3
```

```
desconto=valor * 0.10 ; // forma 4
```

```
novoval=valor-desconto;
```

OPERAÇÃO ARITMÉTICAS

Faça um programa que receba a temperatura em Fahrenheit e converta para Celsius.

Fórmula:

$$\text{Celsius} = ((\text{fahrenheit} - 32) * 5) / 9$$

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

Resolução

```
float cel,fah;
```

```
int main(){
```

```
printf("Digite a temperatura em Fahrenheit\n");
```

```
scanf("%f",&fah);
```

```
cel=((fah-32)*5)/9;
```

```
printf("\nA temperatura em Celsius é: %.2f\n",cel);
```

```
return 0;
```

```
}
```

Teste de Mesa

Dado o algoritmo:

```
float x,y,z;  
int main(){  
    printf("Digite o valor de x\n");  
    scanf("%f",&x);  
    printf("\nDigite o valor de y\n");  
    scanf("%f",&y);  
    z = (x * y) + 2;  
    printf("\nZ é igual a: %.2f\n", z);  
    return 0;  
}
```

Variáveis		
X	Y	Z
3	2	
7	-1	
-2	5	

Teste de Mesa

```
float a,b;  
int main(){  
a=10;  
b=20;  
printf("O valor de b é: %.2f\n",b);  
b=5;  
printf("\nOs valores de a e b são: %.2f %.2f\n",a,b);  
return 0;  
}
```

A<-10
B<-20
Escrever B
B<-5
Escrever A,B

Teste de Mesa

```
float a,b,c;  
int main(){  
a=10;  
b=20;  
c=a;  
b=c;  
a=b;  
printf("O valor de a é: %.2f\n",a);  
printf("\nO valor de b é: %.2f\n",b);  
printf("\n O valor de c é: %.2f\n",c);  
return 0;  
}
```

A<-10
B<-20
C<-A
B<-C
A<-B
Escrever A,B,C

Teste de Mesa

```
float a,b,c;  
int main(){  
a=10;  
b=a+1;  
a=b+1;  
b=a+1;  
printf("O valor de a é: %.2f\n",a);  
a=b+1;  
printf("\nO valor de a é: %.2f\n",a);  
printf("\nO valor de b é: %.2f\n",b);  
return 0;  
}
```

A<-10
B<-A+1
A<-B+1
B<-A+1
Escrever A
A<-B+1
Escrever A,B

OPERAÇÃO ARITMÉTICAS

Escreva um algoritmo para ler um valor e imprimir o seu antecessor e sucessor.

Resolução

algoritmo "calculo"

var n:real

inicio

escreval("Digite um número")

leia(n)

escreval("O antecessor é: ",n-1)

escreval("O sucessor é: ",n+1)

fimalgoritmo

OPERAÇÃO ARITMÉTICAS

Escreva um algoritmo para ler o número total de eleitores de um município, o número de votos brancos, nulos e válidos. Calcular e imprimir o percentual que cada um representa em relação ao total de eleitores.

Fórmula: $\text{total_e} - 100 \%$
 $\text{votos_B} - \text{Pb} \%$

$$\text{Pb} = (\text{votos_B} * 100) / \text{total_e}$$

```
#include<stdio.h>

#include<stdlib.h>

int  num,vb,vn,vv;

float pb,pn,pv;


int main(){

printf("Digite o número total de eleitores\n");

scanf("%d",&num);

printf("\nDigite o número de votos brancos\n");

scanf("%d",&vb);

printf("\nDigite o número de votos nulos\n");

scanf("%d",&vn);

printf("\nDigite o número de votos válidos\n");

scanf("%d",&vv);

pb=(vb*100)/num;

pn=(vn*100)/num;

pv=(vv*100)/num;

printf("O percentual de votos brancos é: %.2f %\n",pb);

printf("O percentual de votos nulos é: %.2f %\n",pn);

printf("O percentual de votos válidos é: %.2f %\n",pv);


system("pause");

return 0; }
```

Resolução

OPERAÇÃO ARITMÉTICAS

O custo de um carro novo ao consumidor é a soma do custo de fábrica com a porcentagem do distribuidor e dos impostos (aplicados ao custo de fábrica).

Supondo que o percentual do distribuidor seja de 28% e os impostos de 45%, escrever um algoritmo para ler o custo de fábrica de um carro, calcular e escrever o custo final ao consumidor.

Resolução

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

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

```
float preco_fi, preco_fa;
```

```
int main(){
```

```
printf("Digite o custo de fábrica do carro\n");
```

```
scanf("%f",&preco_fa);
```

```
preco_fi= preco_fa + (preco_fa *0.28) + (preco_fa *0.45);
```

```
printf("\nO preço final do carro para o consumidor é': %.2f\n",preco_fi);
```

```
system("pause");
```

```
return 0;
```

```
}
```


OPERAÇÃO ARITMÉTICAS

Faça um algoritmo que leia três notas de um aluno, calcule e imprima a média final deste aluno. Considerar que a média é ponderada e que o peso é 2, 3 e 5.

Fórmula: média ponderada = $(n1 * 2 + n2 * 3 + n3 * 5) / 10$

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

```
float n1,n2,n3,media;
```

```
int main(){
printf("Digite a nota 1\n");
scanf("%f",&n1);
printf("Digite a nota 2\n");
scanf("%f",&n2);
printf("Digite a nota 3\n");
scanf("%f",&n3);
```

```
media=(n1*2 + n2*3 + n3*5)/10;
```

```
printf("A média final do aluno é: %.2f\n",media);
```

```
system("pause");
```

```
return 0;
}
```

Resolução

OPERAÇÃO ARITMÉTICAS

Faça um algoritmo que leia um número. Calcule e imprima o:

- O valor do número ao quadrado
- O valor do número ao cubo
- A raiz quadrada do número
- A raiz cúbica do número

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

```
float n,qua,cubo,raiz,raizc,rc;
```

```
int main(){
printf("Digite um número\n");
scanf("%f",&n);
qua=pow(n,2);
cubo=pow(n,3);
raiz=sqrt(n);
raizc=cbrt(n);
rc=pow(n,1.0/3.0); // calcular qualquer outra raiz
printf("\nO número: %.2f ao quadrado é: %.2f\n",n,qua);
printf("\nO número: %.2f ao cubo é: %.2f\n",n,cubo);
printf("\nA raiz do número: %.2f é: %.2f\n",n,raiz);
printf("\nA raiz cúbica do número: %.2f é: %.2f\n",n,raizc);
printf("\nA raiz cúbica do número: %.2f é: %.2f\n",n,rc);
return 0;
}
```

Resolução

OPERAÇÃO ARITMÉTICAS

Faça um programa que calcule e mostre a área de um círculo.

Sabe-se que: **$\text{Area} \leftarrow 3.1415 * \text{raio}^2$**

$\text{Raio}^2 \leftarrow \text{Raio} * \text{Raio}$

ou

Função $\text{pow}(\text{n}, 2)$

Resolução

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

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

```
float area,raio;
```

```
int main(){
```

```
printf("Digite o raio do círculo\n");
```

```
scanf("%f",&raio);
```

```
area=3.1415*(pow(raio,2));
```

```
printf("\nA área da circunferência é:%.2f\n",area);
```

```
return 0;
```

```
}
```

OPERAÇÃO ARITMÉTICAS

Faça um programa que pergunte ao usuário qual a tabuada que ele deseja saber. Calcule e imprima a tabuada do número digitado pelo usuário, começando em 1 e indo até 10.

Formato: $n \times 1 = \text{resp}$

....

$n \times 10 = \text{resp}$

```
#include<stdio.h>
#include<stdlib.h>
int n;

int main(){
printf("Qual tabuada você deseja saber?\n");
scanf("%d",&n);
printf("\n%d x 1 = %d\n",n,n*1);
printf("\n%d x 2 = %d\n",n,n*2);
printf("\n%d x 3 = %d\n",n,n*3);
printf("\n%d x 4 = %d\n",n,n*4);
printf("\n%d x 5 = %d\n",n,n*5);
printf("\n%d x 6 = %d\n",n,n*6);
printf("\n%d x 7 = %d\n",n,n*7);
printf("\n%d x 8 = %d\n",n,n*8);
printf("\n%d x 9 = %d\n",n,n*9);
printf("\n%d x 10 = %d\n",n,n*10);
system("pause");
return 0;
}
```

Resolução

ESTRUTURA CONDICIONAL - SIMPLES

```
if (condição) {
```

```
    linha de comando
```

```
    linha de comando
```

```
}
```

Obs: as linhas de comando só serão executadas se a condição for verdadeira.

Exemplo:

```
if(a>b){
```

```
    a=a+1;
```

```
}
```

ESTRUTURAS DE CONTROLE – Estrutura Condicional

- **Expressões Lógicas**

== igual

> maior

< menor

>= maior ou igual

<= menor ou igual

!= diferente

Condição simples

Faça um algoritmo para ler um número e verificar se o número é maior do que 10, se for imprime ao usuário: “O número é maior do que 10”.

```
if (condição) {
```

```
    linha de comando
```

```
}
```

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

```
float n;
```

```
int main(){
printf("Digite um número\n");
scanf("%f",&n);
if(n>10){
    printf("\nO número é maior do que 10\n");
}
system("pause");
return 0;
}
```

Resolução

ESTRUTURA CONDICIONAL - COMPOSTA

```
if (condição){  
    linhas de comando  
}  
else{  
    linhas de comando  
}
```

Obs: as linhas de comando só serão executadas se a condição for verdadeira.

Exemplo:

```
if(a==0){  
    printf("A é igual a zero");  
}  
else{  
    printf("A é diferente de zero");  
}
```

Condição composta (if/else)

Ler um valor e escrever se é positivo ou negativo (considere o valor zero como positivo).

```
if (condição){
```

```
    linhas de comando
```

```
}
```

```
else{
```

```
    linhas de comando
```

```
}
```

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

```
float n;
```

```
int main(){
```

```
printf("Digite um número\n");
```

```
scanf("%f",&n);
```

```
if(n>=0){
```

```
    printf("\nO número é positivo\n");
```

```
}
```

```
else{
```

```
    printf("\nO número é negativo\n");
```

```
}
```

```
return 0;
```

```
}
```

Resolução

Condição composta (if/else)

Faça um programa que verifique a validade de uma senha fornecida pelo usuário. A senha é 4531. O programa deve mostrar uma mensagem de permissão de acesso ou não.


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

```
int senha;
```

```
int main(){
```

```
printf("Digite a senha\n");
```

```
scanf("%d",&senha);
```

```
if(senha==4531){
```

```
    printf("\nAcesso liberado\n");
```

```
}
```

```
else{
```

```
    printf("\nAcesso não liberado\n");
```

```
}
```

```
return 0;
```

```
}
```

Resolução

Condição composta (if/else)

Faça um algoritmo que leia as notas $n1$ e $n2$ de um aluno. Calcular a média aritmética e escrever uma mensagem que diga se o aluno foi ou não aprovado.

Obs: Considere que a média para aprovação é 6,0.

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

```
float n1,n2,media;
```

```
int main(){  
    printf("Digite a nota 1\n");  
    scanf("%f",&n1);  
    printf("\nDigite a nota 2\n");  
    scanf("%f",&n2);  
    media=(n1+n2)/2;  
    if(media>=6){  
        printf("Aluno aprovado, nota %.2f\n",media);  
    }  
    else{  
        printf("Aluno reprovado, nota %.2f\n",media);  
    }  
    return 0;  
}
```

Resolução

Condição composta (if/else)

Faça um algoritmo que leia o ano atual e o ano de nascimento de uma pessoa. Escrever uma mensagem que diga se ela poderá ou não votar este ano.

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

```
int ano_n,ano_a,idade;
```

```
int main(){
```

```
printf("Digite o ano que você nasceu\n");
```

```
scanf("%d",&ano_n);
```

```
printf("\nDigite o ano atual\n");
```

```
scanf("%d",&ano_a);
```

```
idade= ano_a - ano_n;
```

```
if(idade>=16){
```

```
printf("\nPode votar este ano,você tem: %d\n",idade);
```

```
}
```

```
else{
```

```
printf("\nNão pode votar este ano,você tem: %d\n",idade);
```

```
}
```

```
return 0;
```

```
}
```

Resolução

Condição composta (if/else)

As maçãs custam R\$1,30 cada se forem compradas menos de uma dúzia, e R\$1,00 se forem compradas pelo menos 12. Escreve um programa que leia o número de maçãs compradas, calcule e escreva o custo total da compra.

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

```
int qde;
float preco;
```

Resolução

```
int main(){
printf("Quantas maçãs você deseja comprar?\n");
scanf("%d",&qde);
if(qde>=12){
    preco=qde*1.00;
    printf("\nO valor total da compra é: %.2f\n",preco);
}
else{
    preco=qde*1.30;
    printf("\nO valor total da compra é: %.2f\n",preco);
}
return 0;
}
```

Condição composta (if/else)

Faça um algoritmo para ler: quantidade atual em estoque, quantidade máxima em estoque e quantidade mínima em estoque de um produto.

Calcular e escrever a quantidade média:

$\text{quantidade média} = (\text{quantidade máxima} + \text{quantidade mínima}) / 2.$

Se a quantidade em estoque for maior ou igual à quantidade média escrever a mensagem 'Não efetuar compra', senão escrever a mensagem 'Efetuar compra'.


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

```
int qde, qde_min, qde_max;  
float qde_med;
```

```
int main(){  
    printf("\nDigite a quantidade atual em estoque de produto\n");  
    scanf("%d",&qde);  
    printf("\nDigite a quantidade mínima em estoque de um produto\n");  
    scanf("%d",&qde_min);  
    printf("\nDigite a quantidade máxima em estoque de um produto\n");  
    scanf("%d",&qde_max);  
    qde_med=(qde_max + qde_min)/2;  
    if(qde>=qde_med){  
        printf("\nNão efetuar compra\n");  
    }  
    else{  
        printf("\nEfetuar compra\n");  
    }  
    return 0;  
}
```

Resolução

Condição composta (if/else)

Faça um algoritmo que leia um número inteiro e verifique e mostre se ele é par ou ímpar.

Dicas:

- Número par: quando dividido por 2 produz resto 0.
- Utilize o operado % para pegar o resto da divisão.

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

```
int num;
```

Resolução

```
int main(){
```

```
    printf("Digite um número\n");
```

```
    scanf("%d",&num);
```

```
    if(num%2==0){
```

```
        printf("\nO numero %d e par\n",num);
```

```
    }
```

```
    else{
```

```
        printf("\nO numero %d e impar\n",num);
```

```
    }
```

```
    return 0;
```

```
}
```

TABELA VERDADE

Operadores lógicos: são utilizados para compararmos duas ou mais condições.

Tabela Verdade para “&&” (e)

1ª && 2ª = resp

V && V = V

V && F = F

F && V = F

F && F = F

Tabela Verdade para “||” (ou)

1ª || 2ª = resp

V || V = V

V || F = V

F || V = V

F || F = F

Condição com operador lógico - &&

Escreva um algoritmo para ler 1 número. Verificar e imprimir se ele está contido no intervalo entre 5 e 100.

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

```
float num;
```

Resolução

```
int main(){
printf("Digite um número\n");
scanf("%f",&num);
    if(num>=5 && num<=100){
        printf("\nO número %.2f está contido no intervalo\n",num);
    }
system("pause");
return 0;
}
```

Condição com operador lógico - ||

Uma empresa quer verificar se o empregado está qualificado para a aposentadoria ou não. Para estar em condições, um dos seguintes requisitos deve ser satisfeito:

- Ter no mínimo 65 anos de idade;
- Ter trabalhado no mínimo 30 anos.
- Ter no mínimo 60 anos e ter trabalhado no mínimo 25 anos.

Com base nas informações acima, construa um programa que leia: a idade e o tempo de trabalho do funcionário. O programa deverá imprimir a idade, o tempo de trabalho e a mensagem “Requer aposentadoria” ou “Não requer aposentadoria”.

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

```
int idade;
float tempo;
```

```
int main (){
```

```
printf("Digite sua idade\n");
scanf("%d",&idade);
printf("Digite seu tempo de trabaho\n");
scanf("%f",&tempo);
```

```
if ((idade>=65) || ( tempo>=30) || ((idade>=60)&&(tempo>=25))){
```

```
    printf("Requer aposentadoria"); }
```

```
else{
```

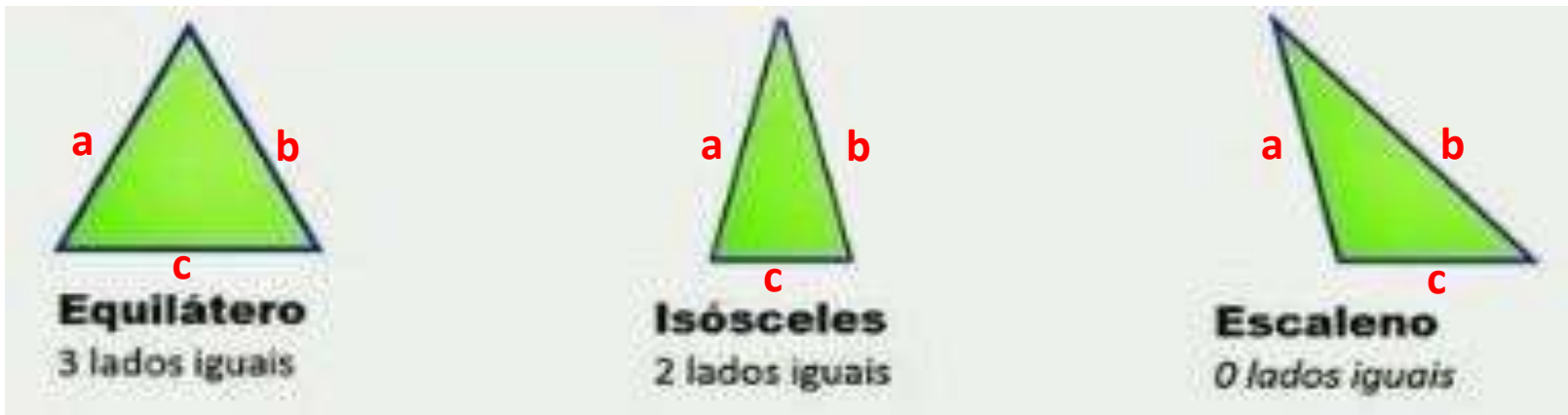
```
    printf("Não requer aposentadoria"); }
```

```
system("pause");
return 0;
}
```

RESOLUÇÃO

ESTRUTURA CONDICIONAL ENCADEADA

Faça um programa que leia os três lados de um triângulo e diga ao usuário qual o tipo do triângulo:



```
#include<stdio.h>
float a,b,c;
int main(){
printf("Digite o tamanho do lado a\n");
scanf("%f",&a);
printf("\nDigite o tamanho do lado b\n");
scanf("%f",&b);
printf("\nDigite o tamanho do lado c\n");
scanf("%f",&c);
if(a+b>c && a+c>b && b+c>a){
    if((a==b)&&(b==c)){
        printf("\nTriângulo equilátero\n");    }
    else if((a==b) || (b==c) || (c==a)){
        printf("\nTriângulo isóceles\n");    }
    else{
        printf("\nTriângulo escaleno\n");    }
}
else{
    printf("\nNão é um triângulo\n");
}
    return 0;
}
```

RESOLUÇÃO

ESTRUTURA CONDICIONAL ENCADEADA

Escreva um algoritmo que receba duas notas de um aluno e calcule e imprima o valor da sua média aritmética e se está aprovado, reprovado e ou de exame.

Critério de aprovação: ≥ 7 e ≤ 10

Critério para exame: ≥ 3 e < 7

Critério de reprovação: ≥ 0 e < 3

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

```
float n1,n2,media;
```

```
int main(){
```

```
printf("Digite as notas 1 e 2\n");
```

```
scanf("%f %f",&n1,&n2);
```

RESOLUÇÃO

```
if(n1>=0 && n1<=10 && n2>=0 && n2<=10){
```

```
media=(n1+n2)/2;
```

```
if(media>=7){
```

```
printf("\nAprovado com média: %.2f\n",media); }
```

```
else if(media>=3 && media<7){
```

```
printf("\nExame com média: %.2f\n",media); }
```

```
else{
```

```
printf("\nReprovado com média: %.2f\n",media); }
```

```
}
```

```
else
```

```
printf("\nNota Invalida\n");
```

```
return 0;
```

```
}
```

ESTRUTURA CONDICIONAL

Faça um programa para ler a quantidade adquirida de um produto e o preço unitário.

- **Calcular e escrever o total:**

$\text{total} = \text{quantidade adquirida} * \text{preço unitário}$

- **O desconto:**

- se quantidade ≤ 5 o desconto será de 2%

- se quantidade estiver entre 6 e 10 o desconto será de 3%

- se quantidade > 10 o desconto será de 5%

- **Total a pagar:**

$\text{total a pagar} = \text{total} - \text{desconto}$

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

```
float preco,total,desc;
```

```
int qde;
```

```
int main(){
```

```
printf("Quantidade do produto:\n");
```

```
scanf("%d",&qde);
```

```
printf("\nPreço: \n");
```

```
scanf("%f",&preco);
```

```
if(preco>0 && qde>0){
```

```
    total=qde*preco;
```

```
    if(qde<=5){
```

```
        desc=total*0.02;
```

```
    }
```

```
    else if (qde>5 && qde<=10){
```

```
        desc=total*0.03;
```

```
    }
```

```
    else{
```

```
        desc=total*0.05;
```

```
    }
```

```
    printf("\nTotal: R$ %.2f\n",total);
```

```
    printf("\nDesconto: R$ %.2f\n",desc);
```

```
    printf("\nTotal a pagar: R$ %.2f\n",total-desc);
```

```
}//if
```

```
else
```

```
    printf("Preço inválido ou quantidade inválida\n");
```

```
    return 0;
```

```
}
```

RESOLUÇÃO

ESTRUTURA CONDICIONAL COMPOSTA

Faça um programa que receba a idade de um nadador e mostre a sua categoria usando as regras a seguir:

Categoria	Idade
Infantil	5 a 7
Juvenil	8 a 10
Adolescente	11 a 15
Adulto	16 a 30
Sênior	Acima de 30

```
#include<stdio.h>
int idade;
int main(){
printf("Digite sua idade:\n");
scanf("%d",&idade);
if(idade>=5 && idade<=130){
if(idade<=7){
    printf("Categoria Infantil\n");    }
    else if(idade>=8 && idade<=10){
        printf("Categoria Juvenil\n");    }
        else if(idade>=11 && idade<=15){
            printf("Categoria Adolescente\n");    }
            else if(idade>=16 && idade<=30){
                printf("Categoria Adulto\n");    }
                else{
                    printf("Categoria Senior\n");    }
        }
    else{
        printf("\nIdade fora do intervalo\n");
    }
    return 0;
}
```

Resolução

ESTRUTURA CONDICIONAL

Faça um algoritmo que leia dois números (considere que não serão lidos valores iguais) e escrever o maior deles.

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

```
int n1,n2;
```

```
int main(){
```

```
printf("Digite um numero\n");
```

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

```
printf("\nDigite outro número\n");
```

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

```
if(n1!=n2){
```

```
if(n1>n2){
```

```
    printf("\nO número %d é maior do que o número %d\n",n1,n2);
```

```
}
```

```
else{
```

```
    printf("\nO número %d é maior do que o número %d\n",n2,n1);
```

```
}
```

```
}
```

```
else{
```

```
printf("\nDigite valores diferentes\n");
```

```
}
```

```
return 0; }
```

Resolução

ESTRUTURA CONDICIONAL

Uma agência bancária possui dois tipos de investimentos, conforme o quadro a seguir. Faça um programa que receba o tipo de investimento e o valor do investimento e que calcule e mostre o valor corrigido de acordo com o tipo de investimento.

Tipo	Descrição	Rendimento Mensal
1	Poupança	3%
2	Fundo de renda fixa	4%

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

```
int cod;
```

```
float valor;
```

Resolução

```
int main(){
```

```
printf("Digite o tipo de investimento: 1-Poupança e 2-Fundo de renda fixa\n");
```

```
scanf("%d",&cod);
```

```
if(cod==1 || cod==2){
```

```
    printf("Qual valor você deseja investir?\n");
```

```
    scanf("%f",&valor);
```

```
    if(valor>0){
```

```
        if(cod==1){
```

```
            printf("\nO valor corrigido é: R$ %.2f\n", valor*1.03); }
```

```
        else{
```

```
            printf("\nO valor corrigido é: R$ %.2f\n", valor*1.04); }
```

```
        }//if2
```

```
        else
```

```
            printf("\nValor inválido\n");
```

```
        }//if1
```

```
    else
```

```
        printf("\nCódigo inválido\n");
```

```
    return 0;
```

```
}
```

ESTRUTURA CONDICIONAL

Calcula o peso ideal – lista de exercícios

Resolução

```
#include<stdio.h>
#include<ctype.h>
char genero;
float alt;
int main(){
printf("Qual seu genero: M - masculino ou F- Feminino\n");
scanf(" %c",&genero);
genero=toupper(genero); //tolower - minusculo
if(genero=='M' || genero=='F'){
    printf("\nQual sua altura?\n");
    scanf("%f",&alt);
    if(alt>0){
        if (genero=='M'){
            printf("\nSeu peso ideal e %.2f\n",(72.7*alt)-58); }
        else{
            printf("\nSeu peso ideal e %.2f\n",(62.1*alt)-44.7); }
        }//if2
    else
        printf("\nAltura Invalida\n");
} //if1
else
    printf("\nGenero invalido\n");
return 0;
}
```

Resolução

```
#include<stdio.h>
#include<string.h>
float altura,peso;
char ge[10];

int main(){
printf("Digite sua altura\n");
scanf("%f",&altura);
printf("Digite seu genero\n");
scanf("%s",ge);
if (strcmp(ge,"MASCULINO") == 0){
    peso = (72.7 * altura) - 58;
    printf("Peso ideal: %.2f\n",peso);
}
else if (strcmp(ge,"FEMININO") == 0) {
    peso = (62.1 * altura) - 44.7;
    printf("Peso ideal: %.2f\n",peso);
}
else
    printf("\nGenero invalido\n");
return 0;
}
```

SWITCH/CASE

Faça um programa que simule uma calculadora simples, onde o usuário digita dois números e o tipo de operação aritmética que ele deseja realizar: soma, subtração, multiplicação ou divisão.


```
#include<stdio.h>

float num1,num2;
char op;

int main(){
printf ("Digite o número 1:\n");
scanf ("%f",&num1);
printf ("\nDigite o operador: + - * /\n");
scanf (" %c",&op);
printf ("\nDigite o número 2:\n");
scanf ("%f",&num2);
```

```
switch (op){
```

```
case '+':
printf("\nO resultado da soma é: %.2f\n",num1+num2);
break;
case '-':
printf("\nO resultado da subtração é: %.2f\n",num1-num2);
break;
case '*':
printf("\nO resultado da multiplicação é: %.2f\n",num1*num2);
break;
```

```
case '/':
if(num2!=0)
printf("\nO resultado da divisão é: %.2f\n",num1/num2);
else
printf("\nNão existe divisão por 0\n");
break;
default:
printf("\nOperador invalidon");
} //switch
return 0;
}
```

RESOLUÇÃO

SWITCH/CASE

Faça um programa verifique e mostre o intervalo em que o número está contido.

Intervalo:

- -10 à 0
- 1 à 11
- 12 à 24
- numero igual 25

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

```
int n;
```

```
int main(){
```

```
printf("Digite um numero\n");
```

```
scanf("%d",&n);
```

```
if(n>=-10 && n<=0){
```

```
    printf("\nNumero esta no intervalo entre -10 e 0\n");
```

```
}
```

```
else if(n>=1 && n<=11){
```

```
    printf("\nNumero esta no intervalo entre 1 e 11\n");
```

```
}
```

```
    else if(n>=12 && n<=24){
```

```
        printf("\nNumero esta no intervalo entre 12 e 24\n");
```

```
}
```

```
else if(n==25){
```

```
    printf("\nNumero e igual a 25\n");
```

```
}
```

```
else{
```

```
    printf("\nValor invalido\n");
```

```
}
```

```
return 0;
```

```
}
```

RESOLUÇÃO

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

```
int n;
```

```
int main(){
```

```
printf("Digite um numero\n");
```

```
scanf("%d",&n);
```

```
switch(n){
```

```
case -10 ... 0:
```

```
    printf("\nNumero esta no intervalo entre -10 e 0\n");
```

```
break;
```

```
case 1 ... 11:
```

```
    printf("\nNumero esta no intervalo entre 1 e 11\n");
```

```
break;
```

```
case 12 ... 24:
```

```
    printf("\nNumero esta no intervalo entre 12 e 24\n");
```

```
break;
```

```
case 25:
```

```
    printf("\nNumero e igual a 25\n");
```

```
break;
```

```
default:
```

```
    printf("\nValor invalido\n");
```

```
} //switch
```

```
return 0; }
```

RESOLUÇÃO

SWITCH/CASE

Faça um programa que peça ao usuário que digite uma letra minúscula, verifique e mostre se essa letra é uma vogal ou uma consoante.

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

```
#include<ctype.h>
```

```
char letra;
```

```
int main(){
```

```
printf("Digite uma letra\n");
```

```
scanf(" %c",&letra);
```

```
letra=tolower(letra); //letra minúscula
```

```
if (isalpha(letra)){ // verifica se é uma letra do alfabeto
```

```
    switch(letra){
```

```
        case 'a':
```

```
        case 'e':
```

```
        case 'i':
```

```
        case 'o':
```

```
        case 'u':
```

```
            printf("\nA letra %c e uma vogal\n",letra);
```

```
        break;
```

```
        default:
```

```
            printf("\nA letra %c e uma consoante\n",letra);
```

```
    }
```

```
}
```

```
else
```

```
    printf("\nNao e letra\n");
```

```
return 0;
```

```
}
```

RESOLUÇÃO

SWITCH/CASE

Dado o valor do produto e a forma de pagamento.

1= à vista;

2= à prazo.

Se o produto for pago à vista aplique um desconto de 10% e mostre o valor final, senão informe o mesmo valor do produto.

```
#include<stdio.h>

float valor,total;

int op;

int main() {
    printf("\nInforme o valor do produto:\n");
    scanf("%f",&valor);
    printf("\nForma de pagamento:\n");
    printf("\n1 - A vista ");
    printf("\n2 - A prazo \n");
    printf("\nEscolha a forma de pagamento:\n");
    scanf("%d",&op);
    switch(op){
        case 1:
            total=valor*0.90;
            printf("\nO total a pagar a vista = R$ %.2f",total);
            break;
        case 2:
            printf("\nO total a pagar a prazo = R$ %.2f", valor);
            break;
        default:
            printf("\nOpcao invalida\n");
    }
    return 0; }
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO

Temos três estruturas de repetição usuais nas linguagens de programação:

- **laço contado:** **for**
- **laços condicionados:** **while (condição){ ... }** e **do{ ... } while(condição);**

Laço: for

Esta estrutura repete uma sequência de comandos um determinado número de vezes.

```
for(i=0;i<10;i++){  
    linhas de comandos  
}
```

ou

```
for(i=1;i<=10;i++){  
    linhas de comandos  
}
```

ESTRUTURA DE REPETIÇÃO FOR

Faça um programa que imprima os números de 1 até 100.

```
for(i=0;i<quantidade;i++){
```

```
    linhas de comando
```

```
}
```

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

```
int i;
```

RESOLUÇÃO

```
int main (){
```

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

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

```
}
```

```
return 0;
```

```
}
```

ESTRUTURA DE REPETIÇÃO FOR

Faça um programa que imprima os números de 100 até 1.

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

```
int i;
```

RESOLUÇÃO

```
int main (){
```

```
for (i=100;i>=1;i--){
```

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

```
}
```

```
return 0;
```

```
}
```

ESTRUTURA DE REPETIÇÃO FOR

A conversão de graus Fahrenheit para Celsius é obtida por $c = 0.56 * (f - 32)$. Faça um programa que calcule e escreva uma tabela de graus Celsius e graus Fahrenheit, cujos graus variem de 50 a 65 de 1 em 1.

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

```
int i;
```

```
float c;
```

RESOLUÇÃO

```
int main(){
```

```
for(i=50;i<=65;i++){
```

```
    c = 0.56 * (i - 32);
```

```
    printf("\nA temperatura em Celsius é: %2f\n",c);
```

```
}
```

```
return 0;
```

```
}
```

ESTRUTURA DE REPETIÇÃO FOR

Escreva um programa que receba um valor **n** positivo e escreva uma contagem regressiva do valor até 0. Seu programa deve testar e só aceitar valores positivos.

Resolução:

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

```
int i,n;
```

```
int main(){
```

```
do{
```

```
printf("Digite um número inteiro\n");
```

```
scanf("%d",&n);
```

```
}
```

```
while(n<=0);
```

```
for(i=n;i>=0;i--){
```

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

```
}
```

```
return 0;
```

```
}
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO FOR

Uma sala de aula tem n alunos que gostariam de calcular o volume de um cilindro circular. Escreva um programa que solicite a cada um destes alunos a altura e o raio de seus respectivos cilindros. Calcule e imprima o valor do volume de cada aluno e verifique e mostre qual foi o maior volume encontrado e de qual aluno pertence.

Use a fórmula:

Volume de um cilindro = $3.141592 * \text{raio} * \text{raio} * \text{altura}$.

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

```
int i,n,idmaior;
```

```
float raio,volume,altura,maior;
```

```
int main(){
```

```
printf("Quantos alunos tem na sala?\n");
```

```
scanf("%d",&n);
```

```
for(i=1;i<=n;i++){
```

```
printf("\nDigite a altura do cilindro\n");
```

```
scanf("%f",&altura);
```

```
printf("\nDigite o raio do cilindro\n");
```

```
scanf("%f",&raio);
```

```
volume= 3.141592 * raio * raio * altura;
```

```
printf("\nO volume do seu cilindro é: %.2f\n",volume);
```

```
if(volume>maior){
```

```
    maior=volume;
```

```
    idmaior=i;
```

```
}
```

```
}
```

```
printf("\nO maior volume de cilindro encontra foi: %.2f do aluno: %d\n",maior,idmaior);
```

```
return 0;
```

```
}
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO FOR - CONTADOR

Faça um programa que leia a altura de 10 pessoas e mostre a quantidade de pessoas com altura maior do que 1.60.

Contador de inteiros:

`cont=cont+1` ou `cont++`

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

```
int i, maior=0;
```

```
double altura;
```

```
int main (){
```

```
for (i=0;i<3;i++){
```

```
    printf("Digite sua altura\n");
```

```
    scanf("%lf", &altura);
```

```
        if(altura>1.60)
```

```
            maior++;
```

```
}
```

```
printf("\n Entre as 10 pessoas %d tem mais de 1.60 \n", maior);
```

```
return 0;
```

```
}
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO FOR - CONTADOR

Faça um programa que verifique e mostre quantos números entre 2000 e 4000 são pares e quantos são ímpares.

Dica:

Número par: quando dividido por 2, produz o resto igual a zero.

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

```
int i, par, impar;
```

```
int main (){
```

```
for (i=2000;i<2010;i++){
```

```
    if((i % 2)==0)
```

```
        par++;
```

```
    else
```

```
        impar++;
```

```
}
```

```
printf("\n A quantidade de números pares é %d \n", par);
```

```
printf("\n A quantidade de números impares é %d \n", impar);
```

```
return 0;
```

```
}
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO FOR - CONTADOR

Faça um programa que leia o preço que 10 pessoas pagaram pelo mesmo produto x e mostre a quantidade de pessoas que pagaram menos do que R\$ 80,00.


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

```
int i,cont;
```

```
float preco;
```

```
int main(){
```

```
for(i=0;i<10;i++){
```

```
    printf("Digite o preço pago pelo produto\n");
```

```
    scanf("%f",&preco);
```

```
    if(preco<80)
```

```
        cont++;
```

```
}
```

```
printf("\nHá %d pessoas que pagaram menos de R$80,00 no produto X\n",cont);
```

```
return 0;
```

```
}
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO FOR - CONTADOR

Escreva um algoritmo para ler 30 números. Verificar e imprimir quantos estão no intervalo entre 5 e 100.

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

```
int i,cont;
```

```
float num;
```

```
int main(){
```

```
for (i=0;i<5;i++){
```

```
    printf("Digite um número\n");
```

```
    scanf("%f",&num);
```

```
    if(num>5 && num<100)
```

```
        cont++;
```

```
}
```

```
printf("\nHá %d números no intervalo\n",cont);
```

```
return 0;
```

```
}
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO FOR - CONTADOR

Escreva um algoritmo que calcule a média aritmética de 10 alunos, e verifique e mostre quantos estão aprovados, reprovado e de exame.

Fórmula:

Cálculo da média: $(n1+n2)/2$

Intervalos:

Critério de aprovação: ≥ 7 e ≤ 10

Critério para exame: ≥ 3 e < 7

Critério de reprovação: ≥ 0 e < 3

```

#include <stdio.h>
int i, conta, conte, contr;
float n1, n2, media;
int main(){
    for (i=0; i<10; i++){
        printf("\ndigite sua primeira nota\n");
        scanf("%f", &n1);
        printf("\ndigite sua segunda nota\n");
        scanf("%f", &n2);
        if(n1>=0 && n1<=10 && n2>=0 && n2<=10){
            media=(n1+n2)/2;
            if(media>=7)
                conta++;
            else if (media>=3 && media<7)
                conte++;
            else
                contr++;
        }
        else{
            printf("\nNota invalida\n");
            i--;
        }
    }
    printf("\no numero de alunos aprovados e %d\n", conta);
    printf("\no numero de alunos em exame e %d\n", conte);
    printf("\no numero de alunos reprovados e %d\n", contr);
    return 0;
}

```

RESOLUÇÃO

Exercício laço

Uma firma fez uma pesquisa de mercado para saber se as pessoas gostaram ou não de um novo produto lançado no mercado. Para isso forneceu o sexo do entrevistado e sua resposta (S- Sim ou N- Não). Sabe-se que foram entrevistadas dez pessoas. Faça um programa que calcule e mostre:

- o número de pessoas que respondeu sim;
- o número de pessoas que respondeu não;
- o número de mulheres que respondeu sim;
- a percentagem de homens que respondeu não entre todos os homens analisados.

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

int i,simm,nao,mulher,homem,totalh;
char sexo,resp;
float perc;

int main(){
for(i=0;i<3;i++){
printf("Qual seu sexo: M - masculino ou F- Feminino\n");
scanf("%s",&sexo);
sexo=toupper(sexo); //tolower - minusculo
```

```
if(sexo=='M' | | sexo=='F'){
    printf("\nVocê gostou do novo produto?\n");
    scanf("%s",&resp);
    resp=toupper(resp);
if (resp=='N')
        nao++;
else
        simm++;
if(sexo=='F'&& resp=='S')
        mulher++;
if(sexo=='M')
        totalh++;
if(sexo=='M' && resp=='N')
        homem++;
}// if sexo
else
    printf("\nSexo inválido\n");
}//for
```

```
perc=(homem*100)/totalh;
```

```
printf("\nO número de pessoas que respondeu sim: %d\n",simm);
printf("\nO número de pessoas que respondeu não: %d\n",nao);
printf("\nO número de mulheres que respondeu sim: %d\n",mulher);
printf("\nO percentual de homens que respondeu não: %.2f\n\n",perc);
```

```
return 0;
}
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO FOR - ACUMULADOR

Faça um programa para calcular a altura média de uma sala com 30 alunos.

Acumulador:

```
acum= acum +variável;
```

ou

```
acum+=variável;
```



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

```
int i;
```

```
double acum,altura,media;
```

```
int main (){
```

```
for (i=0;i<3;i++){
```

```
    printf("Digite sua altura\n");
```

```
    scanf("%lf", &altura);
```

```
    acum+=altura;
```

```
}
```

```
media= acum/3;
```

```
printf("\n A altura média é: %2.2lf \n", media);
```

```
return 0;
```

```
}
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO FOR - ACUMULADOR

Faça um algoritmo par ler o código e o preço de 15 produtos, calcular e escrever:

- a) o maior preço lido e de qual produto é.
- b) a média aritmética dos preços dos produtos.

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

```
int i,idmaior;
```

```
double preco,maior,acum,med;
```

```
int main(){
```

```
for(i=1;i<16;i++){
```

```
printf("Digite o preço do produto\n");
```

```
scanf("%lf",&preco);
```

```
acum+=preco;
```

```
if(preco>maior){
```

```
    maior=preco;
```

```
    idmaior=i;
```

```
}
```

```
}
```

```
med=acum/15;
```

```
printf("\nA média aritmética dos preços dos produtos é: %.2lf\n",med);
```

```
printf("\nO maior preço encontrado foi: %.2lf do produto %d\n\n",maior,idmaior);
```

```
return 0;
```

```
}
```

RESOLUÇÃO

ESTRUTURA DE REPETIÇÃO FOR - ACUMULADOR

Escreva um algoritmo para ler 20 números. Todos os números lidos com valor inferior a 50 devem ser somados. Escreva o valor final da soma efetuada e a média aritmética de todos os números.

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

```
int i;
```

```
double acum,n,med,acumed;
```

```
int main(){
```

```
for(i=0;i<20;i++){
```

```
    printf("Digite um número:\n");
```

```
    scanf("%lf",&n);
```

```
    acumed+=n;
```

```
    if(n<50)
```

```
        acum+=n;
```

```
}
```

```
med=acumed/20;
```

```
printf("\nA média dos números é: %.2lf\n",med);
```

```
printf("\nA somatória dos números menores do que 50 é: %.2lf\n\n",acum);
```

```
return 0;
```

```
}
```

RESOLUÇÃO

Laço: while (condição){ .. }

Faça um algoritmo que receba n números. O programa deve fornecer como saída a soma dos números positivos.

Estrutura:

```
while (condição){
```

```
    linhas de comando
```

```
    incrementa contador ex: i++;
```

```
}
```

```
#include<stdio.h>
float num,acum=0;
int n,i;
```

```
int main(){
printf("Quantos números você deseja somar?\n");
scanf("%d",&n);
while(i<n){
printf("Digite um número\n");
scanf("%f",&num);
if(num>=0)
    acum+=num;
i++;
}
printf("\nA somatória dos números é: %.2f\n",acum);
return 0;
}
```

RESOLUÇÃO

Estrutura De Repetição - While

Uma empresa com 25 funcionários decide dar aumento de 30% aos funcionários com salários inferiores a R\$ 1000,00. Faça um algoritmo que receba o salário de cada um dos funcionários e mostre o valor do salário reajustado ou uma mensagem, caso o funcionário não tenha direito ao aumento.


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

```
float sal,nsal;
```

```
int i;
```

```
int main(){
```

```
while(i<26){
```

```
printf("Digite seu salário\n");
```

```
scanf("%f",&sal);
```

```
if(sal<1000){
```

```
    nsal=sal*1.3;
```

```
    printf("\nSeu novo salário é: %.2f\n",nsal);
```

```
}
```

```
else
```

```
    printf("\nVocê não tem direito a aumento\n");
```

```
i++;
```

```
}
```

```
return 0;
```

```
}
```

RESOLUÇÃO

Estrutura de Repetição - While

A prefeitura de uma cidade deseja fazer uma pesquisa entre seus habitantes, para tal, será utilizada uma amostra de 100 pessoas.

Faça um algoritmo para coletar dados sobre o salário e número de filhos de cada habitante e após as leituras, imprimir:

- a) média de salário da população;
- b) média do número de filhos;
- c) quantidade de pessoas com salário menor que \$150,00.

```
#include<stdio.h>

int i=0,nf,cont;

double sal,acums,acum,meds,medf;


int main(){
while(i<3){
printf("Digite seu salário\n");
scanf("%lf",&sal);

printf("Digite o número de filhos\n");
scanf("%d",&nf);

acums+=sal;
acum+=nf;
if(sal<1000)
    cont++;
    i++;
}

meds=acums/3;
medf=acum/3;

printf("\nA média aritmética dos salários é: %.2lf\n",meds);
printf("\nA média aritmética do n de filhos é: %.2lf\n",medf);
printf("\n%d pessoa(s) recebe(m) menos do que R$1.000,00\n",cont);

return 0;

}
```

RESOLUÇÃO

Estrutura De Repetição - While

Faça um programa que leia n idade e que calcule e mostre:

- a quantidade de pessoas com idade maior ou igual a 18 anos.
- a média das idades digitadas.

```
#include<stdio.h>
int i=0,n,id,cont;
float med,acum=0;
```

```
int main(){
printf("Quantos idades você deseja ler?\n");
scanf("%d",&n);
```

```
while(i<n){
printf("\nDigite a idade\n");
scanf("%d",&id);
```

```
acum+=id;
```

```
if(id>=18)
```

```
    cont++;
```

```
i++;
```

```
}
```

```
med=acum/n;
```

```
printf("\n%d pessoas são maiores de idade\n",cont);
```

```
printf("\nA média das idades é: %.2f\n",med);
```

```
return 0;
```

```
}
```

RESOLUÇÃO

Estrutura De Repetição – While e switch-case

Faça um programa de conversão de base numérica. O programa deverá apresentar uma tela de entrada com as seguintes opções:

< Conversão de base >

- 1: decimal para hexadecimal
- 2: hexadecimal para decimal
- 3: decimal para octal
- 4: octal para decimal
- 5: Encerra

Lembre que, você deve informar as opções para que a partir da opção escolhida, o programa deva pedir o número na base escolhida, lê-lo e apresentá-lo na base desejada. Em seguida, o programa deve perguntar ao usuário se ele deseja retornar menu principal ou finalizar o programa. O problema pode ser estendido a outras de acordo com o interesse do aluno.

```

#include<stdio.h>

int main() {
    int valor, opcao = 0;
    char opcao_c;
    while(opcao!=5) { // se opção igual a 5 encerra o programa
        printf("\n\n\t< Conversao de base >");
        printf("\n 1: decimal para hexadecimal");
        printf("\n 2: hexadecimal para decimal");
        printf("\n 3: decimal para octal");
        printf("\n 4: octal para decimal");
        printf("\n 5: Encerra");
        printf("\n\t\t Informe sua opcao: ");
        scanf("%d",&opcao);
        if ((opcao > 4) || (opcao < 1))
            continue; //se a opção não está entre 1 e 4 voltar no inicio
        printf("\nEntre com o numero para converter: ");
        switch (opcao) {
            case 1:
                scanf("%d", &valor);
                printf("Numero convertido: %x", valor);
                break;
            case 2:
                scanf("%x", &valor);
                printf("Numero convertido: %d", valor);
                break;

```

```

            case 3:
                scanf("%d", &valor);
                printf("Numero convertido: %o", valor);
                break;
            case 4:
                scanf("%o", &valor);
                printf("Numero convertido: %d", valor);
                break;
        } //swicth
    } //while
    return 0;
}

```

Laço do.. while(condição)

O cardápio de uma lanchonete é o seguinte:

Especificação	Código	Preço
Cachorro quente	100	R\$ 10.00
Refrigerante	101	R\$ 5.00

Faça um programa que leia o código dos itens pedidos e as quantidades desejadas. Calcule e mostre o valor a ser pago por item (preço*quantidade) e o total geral do produto. Considere que o cliente deve informar quando o pedido deve ser encerrado.

Estrutura:

```
do{
```

```
    linhas de comando
```

```
    pergunta para saber se deseja continuar ou incrementa contador
```

```
}
```

```
while(condição);
```



```
#include<stdio.h>

float valor,vtotal;
int i,codigo,qtd;
char resp;

int main(){
do{
printf("Digite o código do produto\n");
scanf("%d",&codigo);
printf("Digite a quantidade\n");
scanf("%d",&qtd);
if(codigo==100){
    printf("Cachorro quente");
    valor=qtd*10.0;  }
else if(codigo==101){
    printf("Refrigerante");
    valor=qtd*5.00;    }
printf("\nValor do produto %2.2f\n", valor);
vtotal+=valor;
printf("\nDeseja mais algum produto?\n");
scanf(" %c",&resp);
}
while(resp=='S');
printf("\nO valor total da compra é: %2.2f\n",vtotal);
return 0;
}
```

RESOLUÇÃO

Exercício laço – do..while

O professor de um colégio precisa calcular as médias finais de seus alunos, mas ele não se lembra exatamente quantos alunos ele tem. Para tal, faça um programa que receba as 2 notas de cada dos seus alunos, calcule e mostre a média aritmética e a mensagem que segue a tabela abaixo.

Média Aritmética	Mensagem
0,0 – 3,0	Reprovado
3,0 – 7,0	Exame
7,0 – 10,0	Aprovado

O programa deve terminar quando o professor disser que não há mais alunos para calcular a média.

(usar o laço do ... while(resp== "S"))

```
#include<stdio.h>

double n1,n2,media;
char resp;

int main(){
do{
printf("Digite as notas 1 e 2\n");
scanf("%lf %lf",&n1,&n2);
media=(n1+n2)/2;
if(media>=7 && media<=10)
    printf("\nAprovado, média: %.2f\n",media);
    else if(media>=3 && media<7)
        printf("\nExame, média: %.2f\n",media);
        else if(media>=0 && media<3)
            printf("\nReprovado, média: %.2f\n",media);
        else
            printf("\nNota Invalida\n");
printf("\nDeseja calcular mais alguma nota?\n");
scanf("%s",&resp);
} // do
while(resp == 'S');
return 0;
}
```

RESOLUÇÃO

Vetor

Tipo de variável capaz de armazenar n dados do mesmo tipo.

- Quando usar?

Quando precisamos armazenar mais de um dado do mesmo tipo.

Ex: armazenar 10 idades, 3 salários, 1000 nomes, 50 respostas, dentre outros.

- Como declarar um vetor:

tipo nome[tamanho];

Acesso a cada elemento é feito através de indexação da variável.

– Exemplo:

`v[0] = 0; // acessa o primeiro elemento de v`

...

`v[9] = 9; // acessa o último elemento de v`

`v[10] = 10 // ERRADO (invasão de memória)`

0	1	2	3	4	5	6	7	8	9
15	37	19	73	81	24	66	92	17	22

Vetor

- Como inicializar um vetor:

```
laço: for(i=0;i<=tamanho;i++);  
      printf("Digite um valor\n");  
      scanf("%d",&nome[i]);
```

- Como imprimir um vetor:

```
laço: for(i=0;i<=tamanho;i++);  
      printf("\nPosição: %d do vetor: %d\n",i,nome[i]);
```

Vetor

Faça um programa para criar um vetor de números inteiros com 20 posições. Preencha e imprima o vetor.

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

```
int i,v1[20];
```

```
int main (){
```

```
for (i=0;i<20;i++){
```

```
    printf("Digite um valor \n");
```

```
    scanf("%d", &v1[i]);
```

```
}
```

```
for (i=0;i<20;i++){
```

```
    printf("\n %d", v1[i]);
```

```
}
```

```
return 0;
```

```
}
```

RESOLUÇÃO

Vetor

Escreva um programa que receba a altura e peso de 5 pessoas. Calcule e armazene os valores de IMC de cada pessoa em um vetor chamado `imc`, ao final imprima os IMC das seguintes pessoas: 1ª, 3ª e 5ª.

O IMC é calculado da seguinte forma:

$$\text{IMC} = \text{peso} / \text{altura}^2$$


```
#include<stdio.h>

int i;
float peso[5], alt[5], imc[5];


int main(){
for (i=1; i<=5; i++){
printf("Digite seu peso:\n\n");
scanf("%f",&peso[i]);
printf("Digite seu altura:\n\n");
scanf("%f",&alt[i]);


imc[i]=peso[i]/(alt[i]*alt[i]);
}
printf("\nO IMC da 1ª pessoa é: %2.2f\n",imc[1]);
printf("\nO IMC da 3ª pessoa é: %2.2f\n",imc[3]);
printf("\nO IMC da 5ª pessoa é: %2.2f\n",imc[5]);


return 0;
}
```

Vetor

Construa um vetor de caracteres com 5 posições e armazene os seguintes valores: avião, gato, papagaio, casa, madeira.

Preencher automaticamente

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

```
char v[5][10]={"aviao","gato","papagaio","casa","madeira"};
```

```
int i;
```

```
int main(){
```

```
for(i=0;i<5;i++){
```

```
    printf("%s\n",v[i]);
```

```
}
```

```
return 0;
```

```
}
```

Preencher manualmente

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

```
char v[5][10];
```

```
int i;
```

```
int main(){
```

```
for(i=0;i<5;i++){
```

```
    printf("Digite a palavra\n");
```

```
    scanf(" %s",&v[i]);
```

```
}
```

```
for(i=0;i<5;i++){
```

```
    printf("%s\n",v[i]);
```

```
}
```

```
return 0;
```

```
}
```

Vetor

Construa um programa que preencha aleatoriamente um vetor de inteiros de 5 posições. E imprima a somatória de todos os valores contidos no vetor.

Obs:

```
vetor[i]=rand()%100
```

Gera números aleatórios entre 0 e 100

```
srand(time(NULL));
```

Faz com que os números aleatórios sejam diferentes a cada execução

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

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

```
#include<time.h>
```

```
int Vetor[5],acum,i;
```

```
int main() {
```

```
    srand(time(NULL));
```

```
    for (i=0;i<5;i++){
```

```
        Vetor[i]=rand()%100;
```

```
        printf("Valor no Vetor[%d]é %d\n",i,Vetor[i]);
```

```
        acum=acum+Vetor[i];
```

```
    }
```

```
    printf("\n A soma dos valores contidos no vetor é de %d\n",acum);
```

```
    return 0;
```

```
}
```

Vetor

Construa um algoritmo que carregue 2 vetores com 4 posições e crie um terceiro vetor intercalando o primeiro e o segundo vetor.

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

```
int i,j, v1[4], v2[4], v3[8];
```

```
int main (){
```

```
for (i=0;i<4;i++){
```

```
printf("Digite um valor para o vetor 1 e 2\n");
```

```
scanf("%d %d", &v1[i], &v2[i]);
```

```
v3[j]=v1[i];
```

```
v3[j+1]=v2[i];
```

```
j+=2;
```

```
}
```

```
for (i=0;i<8;i++){
```

```
printf("\n %d\n", v3[i]);
```

```
}
```

```
return 0;
```

```
}
```


Vetor

Faça um programa que preencha 1 vetor de inteiros com 5 posições, e calcule e imprima um segundo vetor que é resultado do incremento de 2 do primeiro vetor.

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

```
int i;
```

```
float a[5],b[5];
```

```
int main (){
```

```
for (i=1;i<=5;i++){
```

```
    printf("\nDigite o valor do primeiro vetor: \n");
```

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

```
    b[i]=a[i]+2;
```

```
}
```

```
for(i=1;i<=5;i++){
```

```
    printf("\n%2.2f\n",b[i]);
```

```
}
```

```
return 0;
```

```
}
```

Exercício ENADE

Exemplo: Considere uma situação em que um professor que queira saber se existem alunos cursando, ao mesmo tempo, as disciplinas A e B, tenha implementado um programa que:

- 1) inicializa um *array* **a** de 30 posições que contém as matrículas dos alunos da disciplina A;
- 2) inicializa outro *array* **b** de 40 posições, que contém as matrículas dos alunos da disciplina B;
- 3) imprime a matrícula dos alunos que estão cursando as disciplinas A e B ao mesmo tempo.

Considere, ainda, que os *arrays* foram declarados e inicializados, não estão necessariamente ordenados, e seus índices variam entre 0 e n - 1, sendo o n o tamanho do *array*.

```
1. for ( i = 0 to 29 ) {  
2.     for ( j = 0 to 39 ) {  
3.  
4.  
5.  
6.     }  
7. }
```

Com base nessas informações, conclui-se que o trecho a ser incluído nas linhas 3, 4 e 5 do código acima, para que o programa funcione corretamente, é:

Exercício ENADE

a)

```
3. if (a[i] == b[j]) {  
4.     print(a[i]);  
5. }
```

b)

```
3. if (a[j] == b[i]) {  
4.     print(a[j]);  
5. }
```

c)

```
3. if (a[i] == b[j]) {  
4.     print(a[j]);  
5. }
```

d)

```
3. if (a[i] == b[i]) {  
4.     print(a[i]);  
5. }
```

e)

```
3. if (a[j] == b[j]) {  
4.     print(a[j]);  
5. }
```

Exercício ENADE

Exemplo: Jogos de tabuleiro são atividades comuns de entretenimento na vida cotidiana das pessoas. Uma de suas características é a necessidade do uso de um tabuleiro com localizações bem definidas para o posicionamento de peças, podendo indicar também as fases do jogo. No livro O Homem que Calculava, de Malba Tahan (São Paulo: Record, 2002, p. 120), há uma história na qual um rei deveria efetuar o pagamento pelos serviços de um de seus conselheiros, dando-lhe uma certa quantidade de grãos de trigo a ser calculada da seguinte forma: coloca-se 1 grão de trigo na primeira casa do tabuleiro, 2 na segunda casa, 4 na terceira casa, 8 na quarta casa e assim dobrando-se sucessivamente até a última casa.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Considerando o tabuleiro 4 x 4 ilustrado acima, contendo a indicação da ordem das casas, construa um único algoritmo que:

- calcule, armazene em uma estrutura e escreva em ordem a quantidade de grãos de trigo em cada casa do tabuleiro;
- calcule, armazene em uma variável e escreva a quantidade total de grãos de trigo presentes no tabuleiro.

Exercício ENADE

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

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

```
int Trigo[16],i,j,TotalTrigo=0;
```

```
int main(){
```

```
a)  for(i=0;i<=15;i++){
      Trigo[i]=pow(2,i);
      TotalTrigo+=Trigo[i];
    }

    for(i=0;i<=15;i++){
      printf("\nQuantidade de grãos de trigo na casa %d é %d\n",i+1,Trigo[i]);
    }

b)  printf("\nO total de grãos de trigo é: %d\n",TotalTrigo);

    return 0;
  }
```

Vetor

Faça um programa que preencha 2 vetores de inteiros com 20 posições cada e calcule e imprima um 3 vetor que é resultado da multiplicação entre o primeiro e o segundo vetor.

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

```
int i;
```

```
float a[20],b[20],c[20];
```

```
int main (){
```

```
for (i=1;i<=20;i++){
```

```
    printf("\nDigite o número 1 \n");
```

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

```
    printf("\nDigite o número 2 \n");
```

```
    scanf("%f",&b[i]);
```

```
    c[i]=a[i]*b[i];
```

```
}
```

```
for(i=1;i<=20;i++){
```

```
    printf("\n%2.2f\n",c[i]);
```

```
}
```

```
return 0;
```

```
}
```


Vetor

Faça um programa que carregue um vetor com 10 números inteiros. Calcule e mostre um vetor resultante ordenado de maneira crescente.

```
#include<stdio.h>

int v[10], i, j, aux;

int main (){
for (i=0;i<9;i++){ //preenche o vetor
    printf("Digite um numero inteiro\n");
    scanf("%d",&v[i]);
}
for (i=0;i<9;i++){ // coloca em ordem crescente
    for(j=0;j<8;j++){
        if(v[j]>v[j+1]){
            aux=v[j];
            v[j]=v[j+1];
            v[j+1]= aux;
        } //if
    } // for interno
} // for externo
for (i=0;i<9;i++){ //imprime
    printf("%d\n",v[i]);
}
return 0;
}
```

Vetor

Construa um algoritmo que faça o cadastro de 100 pacientes de uma clínica, cadastre o nome do paciente, a idade, o sexo, telefone e ao final imprimir os dados dos pacientes.

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

```
int i,tel[2],id[2];
char nome[2][40],sexo[2][40];
```

```
int main (){
for (i=1;i<=2;i++){
    printf("Digite seu nome \n");
    scanf(" %[^\\n]",&nome[i]);
    printf("\\nDigite seu sexo \n");
    scanf("%s",&sexo[i]);
    printf("\\nDigite sua idade \n");
    scanf("%d",&id[i]);
    printf("\\nDigite seu telefone \n");
    scanf("%d",&tel[i]);
}
for (i=1;i<=2;i++){
printf("\\nNome: %s, sexo: %s, idade: %d, telefone: %d\\n",nome[i],sexo[i],id[i],tel[i]);
}
system("pause");
return 0;
}
```

Vetor

Faça um programa que carregue um vetor com oito números inteiros, calcule e mostre dois vetores resultantes:

- O primeiro vetor deve conter os números pares;
- O segundo vetor resultante deve conter os números ímpares;

Cada vetor resultante vai ter no máximo 8 posições, sendo que nem todas as posições obrigatoriamente devem ser utilizada.

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

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

```
int i, num[8], par[8], impar[8],p,im;
```

```
int main(){
```

```
for (i=0; i<8; i++){
```

```
printf("Digite 8 numeros inteiros:\n\n");
```

```
scanf("%d",&num[i]);
```

```
if(num[i]%2 == 0){
```

```
    par[p]=num[i];
```

```
    p++;
```

```
}
```

```
else {
```

```
    impar[im]=num[i];
```

```
    im++;
```

```
}
```

```
}
```

```
printf("\nNumeros pares digitados\n");
```

```
for (i=0; i<p; i++){
```

```
    printf("%d\n",par[i]);
```

```
}
```

```
printf("\nNumeros impares digitados\n");
```

```
for (i=0; i<im; i++){
```

```
    printf("%d\n",impar[i]);
```

```
}
```

```
system("pause");
```

```
return 0;
```

```
}
```

Vetor

Faça um programa que carregue um vetor com quatro números inteiros, calcule e mostre dois vetores resultantes:

- O primeiro vetor deve conter os números positivos;
- O segundo vetor resultante deve conter os números negativos;

Cada vetor resultante vai ter no máximo 4 posições, sendo que nem todas as posições obrigatoriamente devem ser utilizada.

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

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

```
int i, num[4], pos[4], neg[4],p,n;
```

```
int main(){
```

```
for (i=0; i<4; i++){
```

```
printf("Digite 4 numeros inteiros:\n\n");
```

```
scanf("%d",&num[i]);
```

```
if(num[i]>=0){
```

```
    pos[p]=num[i];
```

```
    p++;
```

```
}
```

```
else {
```

```
    neg[n]=num[i];
```

```
    n++;
```

```
}
```

```
}
```

```
printf("\nNumeros positivos digitados\n");
```

```
for (i=0; i<p; i++){
```

```
    printf("%d\n",pos[i]);
```

```
}
```

```
printf("\nNumeros negativos digitados\n");
```

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

```
    printf("%d\n",neg[i]);
```

```
}
```

```
system("pause");
```

```
return 0;
```

```
}
```


Vetor

Um professor aplicou uma prova múltipla escolha em uma sala de aula com 30 alunos e ele precisa corrigir as provas e mostrar a nota de cada um. Para tal, faça um programa que corrija as provas para o professor.

- O professor precisa fornecer o gabarito da prova que contém 5 testes, cada um com 5 alternativas: A, B, C, D e E;
- O programa deve ler as 5 respostas de cada aluno e comparar com a do professor;
- Cada questão que ele acertar ele computa 2 pontos;
- Imprima a nota de todos os alunos.

Dica: declare a biblioteca: **#include<string.h>**

Use o comando **strcmp** para comparar as strings:

if(strcmp(a[i],p[i])==0))

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

```
int i,j,nota[3];
char p[5][10],a[5][10];
```

```
int main(){
printf("Gabarito professor\n");
for (i=1; i<=5; i++){
printf("Digite a resposta da %d pergunta\n",i);
scanf("%s",&p[i]);
}
for(j=1;j<=3;j++){
printf("\nProva do aluno: %d\n",j);
for(i=1;i<=5;i++){
printf("\nDigite a resposta da %d pergunta\n",i);
scanf("%s",&a[i]);

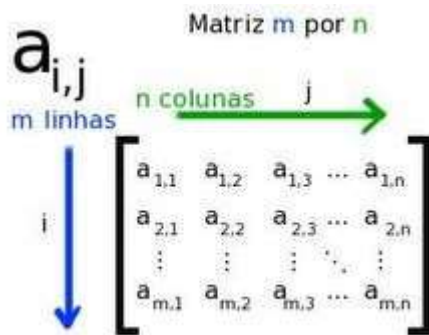
if( strcmp(a[i],p[i])==0){
nota[j]=nota[j]+2;
}
}
}
}
```

```
for(i=1;i<=3;i++){
printf("\nA nota do aluno %d foi de %d\n",i,nota[i]);
}
}
```

```
system("pause");
return 0; }
```

MATRIZ (array bidimensional):

Uma matriz é um array bidimensional, composta por linhas e colunas, como uma tabela.



	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11

```
A[0][0] = 0;  
A[0][1] = 1;  
A[0][2] = 2;  
A[0][3] = 3;  
A[1][0] = 4;  
A[1][1] = 5;  
A[1][2] = 6;  
A[1][3] = 7;  
A[2][0] = 8;  
A[2][1] = 9;  
A[2][2] = 10;  
A[2][3] = 11;
```

Matriz

- Como declarar uma matriz:

tipo nome[linha][coluna];

- Exemplo:

matriz de inteiros 4x4:

int mat[4][4];

Acesso a cada elemento é feito através de indexação de duas variáveis (linha,coluna):

mat[0][0]= 1; // acessa o primeiro elemento da matriz mat

...

mat[3][3] = 16; // acessa o último elemento da matriz mat

- Como preencher uma matriz:

```
laço: for(i=0;i<linha;i++){  
    for(j=0;j<coluna;j++){  
        printf("Digite um valor\n");  
        scanf("%d",&mat[i][j]);  
    } // for interno  
} // for externo
```

- Como imprimir uma matriz :

```
laço: for(i=0;i<linha;i++){  
    for(j=0;j<coluna;j++){  
        printf("[%d][%d]= %d\n",i,j,mat[i][j]);  
    } // for interno  
} // for externo
```

Matriz

Faça um programa para preencher uma matriz 4x4 e imprimir os valores preenchidos com as respectivas posições.

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

```
int mat[4][4],i,j;
```

```
int main(){
```

```
for(i=0;i<4;i++){
```

```
    for(j=0;j<4;j++){
```

```
        printf("Digite o valor para %d e %d\n",i+1,j+1);
```

```
        scanf("%d",&mat[i][j]);
```

```
    }
```

```
}
```

```
for(i=0;i<4;i++){
```

```
    for(j=0;j<4;j++){
```

```
        printf("[%d][%d]= %d\n",i+1,j+1,mat[i][j]);
```

```
    }
```

```
}
```

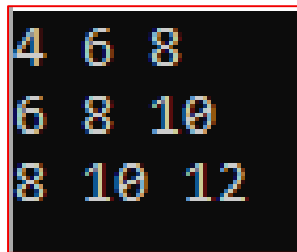
```
return 0;
```

```
}
```

Matriz

Na aula de algoritmos, os alunos desenvolveram uma matriz do tipo 3 x 3, onde $a_{ij} = 2i + 2j$.

Em um determinado momento o computador foi solicitado a mostrar, na tela, a matriz calculada. Faça o algoritmo, onde a matriz a ser mostrada na tela é:



4	6	8
6	8	10
8	10	12


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

```
int a[3][3],i,j;
```

```
int main(){
```

```
for(i=0;i<3;i++){
```

```
    for(j=0;j<3;j++){
```

```
        a[i][j]=(2*(i+1))+(2*(j+1));
```

```
        printf("%d ",a[i][j]);
```

```
    }
```

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

```
}
```

```
return 0;
```

```
}
```

Matriz

Dada uma matriz de dimensão 3x6, calcule a soma de cada coluna dessa matriz e armazene esses valores num vetor. Imprima a matriz e o vetor.

```
#include<stdio.h>
int i,j,M[3][6],V[6];
int main(){
    for(i=0;i<3;i++){
        for(j=0;j<6;j++){
            printf("Digite o valor para %d e %d\n",i+1,j+1);
            scanf("%d",&M[i][j]);
            V[j]=V[j]+M[i][j];
        }
    }
    printf("\n");

    for(i=0;i<3;i++){
        for(j=0;j<6;j++){
            printf("%d",M[i][j]);
        }
        printf("\n");
    }
    for(i=0;i<6;i++){
        printf("\n[%d]= %d\n",i+1,V[i]);
    }
    return 0; }
```

Matriz

Um professor precisa lançar a frequência de uma turma de Linguagem de Programação. O professor possui uma planilha em papel com as presenças (p) e faltas (f) diárias para os 5 alunos da turma nos 3 dias de aula (cada dia de aula corresponde a duas frequências). Faça um programa para auxiliar o professor a obter as informações de que ele necessita. O programa deverá:

a. ler a planilha de frequência para uma matriz de caracteres de 5 x 3 (5 alunos e 3 dias).

b. Escrever o total de faltas para cada um dos 5 alunos, conforme exemplo abaixo.

Exemplo: Aluno n: X faltas

c. Informar o total de alunos reprovados por falta (considere reprovado o aluno que possui mais de 2 faltas).

```
#include <stdio.h>
char planilha[5][3];
int i, j, totalFaltas, totalReprovados;
int main() {
    for (i=0; i<5; i++){ // número de alunos
        totalFaltas=0;
        for (j=0; j<3; j++){ //quantidade de dias
            printf("Aluno %d dia %d\n", i+1, j+1);
            scanf(" %c", &planilha[i][j]);
            if (planilha[i][j]=='f'){
                totalFaltas++;
            }
        }
        printf("Aluno %d teve %d faltas\n", i+1, totalFaltas);
        if (totalFaltas > 2){
            totalReprovados++;
        }
    }
    printf("Total de reprovados por faltas %d\n", totalReprovados);
    return 0;
}
```

Função

O que é?

- É um conjunto de instruções agrupadas em um bloco, que recebe um nome para referenciá-la.

Por quê usar função?

- Divide grandes tarefas em tarefas menores, e permite as pessoas trabalharem sobre o que as outras já fizeram, em vez de partir do nada.
- Para facilitar a leitura do programa de uma forma mais fácil;

Formato Geral de uma Função em C:

```
tipo_da_funcao NomeDaFuncao (Lista_de_Parâmetros)
{
// instruções
}
```

Função

Parâmetros ou Argumentos:

- Este parâmetros possibilitam que se definida sobre quais dados a função deve operar. A função **area(float r)**, por exemplo, recebe como parâmetro o raio da circunferência, permitindo que se calcule a área a partir deste valor.
- Exemplo de uma função para somar 2 números, sendo um float e um inteiro:

```
void soma(float a, int b)  
{  
    float result;  
    result = a+b;  
    printf("A soma de %2.2f com %d é %2.2f\n", a,b,result);  
}
```

Programa completo: função soma + função main

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

void soma(int n1, float n2){//função soma não retorna nada e recebe dois parâmetros
float resultado;
resultado= n1+n2;
printf("A soma de %d com %2.2f é %2.2f\n",n1,n2,resultado);
}

int main(){
int a;
float b;

a=20;
b=15.5;
soma(a,b); // Chamada da função soma(20,15.5);

system("pause");
return 0;
}
```


Programa completo: protótipo + função main + função soma

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

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

```
void soma(int , float); //protótipo da função
```

```
int main(){
```

```
int a;
```

```
float b;
```

```
a=20;
```

```
b=15.5;
```

```
soma(a,b); // Chamada da função soma(20,15.5);
```

```
system("pause");
```

```
return 0;
```

```
}
```

```
void soma(int n1, float n2){ //função soma não retorna nada e recebe dois parâmetros
```

```
float resultado;
```

```
resultado= n1+n2;
```

```
printf("A soma de %d com %2.2f é %2.2f\n",n1,n2,resultado);}
```

Programa completo: protótipo + função main + função soma (retorna valor)

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

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

```
float soma(int , float); //protótipo da função
```

```
int main(){
```

```
int a;
```

```
float b,r;
```

```
a=20;
```

```
b=15.5;
```

```
r=soma(a,b); // Chamada da função soma(20,15.5);
```

```
printf("A soma de %d com %2.2f é %2.2f\n",a,b,r);
```

```
system("pause");
```

```
return 0;
```

```
}
```

```
float soma(int n1, float n2){ //função soma retorna um float e recebe dois parâmetros
```

```
float resultado;
```

```
resultado= n1+n2;
```

```
return resultado;}
```

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

void FUNC1(){
    int B=-100;
    printf("Valor de B dentro da função FUNC1: %d\n", B);}

void FUNC2(){
    int B;
    B = -200;
    printf("Valor de B dentro da função FUNC2: %d\n", B);}

int main(){
    int B;

    B = 10;
    printf("Valor de B: %d\n", B);
    B = 20;
    FUNC1();
    printf("Valor de B: %d\n", B);
    B = 30;
    FUNC2();
    printf("Valor de B: %d\n", B);

    system("pause");
    return 0;}
```

Sobre as variáveis:

- As variáveis valem no bloco que são definidas;
- as variáveis definidas dentro de uma função recebem o nome de **variáveis locais**;
- As variáveis definidas fora de qualquer função são chamadas de **variáveis globais**;
- os parâmetros de uma função valem também somente dentro da função;
- uma variável definida dentro de uma função não é acessível em outras funções, mesmo que estas variáveis tenham nome idênticos.

Função

Calcule a área da esfera, cuja área é: $4\pi r^2$, onde r é o raio e $\pi=3.14$. Construa um algoritmo usando função e utilize o protótipo.

Dica:

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

```
#define PI 3.14
```

```
Area=4*PI*raio*raio
```

```
#include<stdio.h>
#include<stdlib.h>
#define PI 3.14
```

```
float area(float ); //protótipo da função
```

```
int main(){
```

```
float raio,a;
```

```
printf("Digite o raio da esfera\n");
```

```
scanf("%f",&raio);
```

```
a=area(raio); // Chamada da função soma(20,15.5);
```

```
printf("A área da esfera é %2.2f\n",a);
```

```
system("pause");
```

```
return 0;
```

```
}
```

```
float area(float r){ //função soma não retorna nada e recebe dois parâmetros
```

```
float a;
```

```
a= (4*PI*r*r);
```

```
return a;}
```

Função

Transforme a temperatura de Fahrenheit para Celsius usando uma função sem o uso de protótipo.

Fórmula para conversão:

$$\text{Celsius} = (\text{Fahrenheit} - 32) * 5 / 9$$

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

```
float celsius(float fahr){
float c;
c= (fahr-31) * 5/9;
return c;
}
```

```
int main(){
float c,f;
```

```
printf("Digite a temperatura em graus Fahrenheit\n");
scanf("%f",&f);
c=celsius(f); // Chamada da função soma(20,15.5);
printf("Fahrenheit: %2.2f e Celsius: %2.2f\n",f,c);
```

```
system("pause");
return 0;
}
```

Função

Faça um programa que calcule a força necessária para mover um objeto, usando uma função para o cálculo e protótipo.

Fórmula:

Força = massa * aceleração


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

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

```
float forca(float,float);
```

```
int main (){
```

```
float massa, ace, forc;
```

```
printf("Digite a massa \n");
```

```
scanf("%f", &massa);
```

```
printf("Digite a aceleração \n");
```

```
scanf("%f", &ace);
```

```
forc= forca(massa,ace);
```

```
printf("\nA força é: %2.2f\n", forc);
```

```
system("pause");
```

```
return 0;}
```

```
float forca(float m, float a) {
```

```
float f;
```

```
f= m*a;
```

```
return f;}
```

Função

Faça um programa que calcule a área e o perímetro de um retângulo, onde **área = base * altura** e o **perímetro = 2*(base+altura)**. Construa duas funções.

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

```
float area(float,float);
float perimetro(float,float);
```

```
int main(){
float base,alt,a,p;
```

```
printf("Digite o tamanho da base\n");
scanf("%f",&base);
```

```
printf("Digite o tamanho da altura\n");
scanf("%f",&alt);
```

```
a=area(base,alt);
p=perimetro(base,alt);
```

```
printf("A área da retângulo é %2.2f e o perímetro é %2.2f\n",a,p);
```

```
system("pause");
return 0;
}
```

```
float area(float b, float a){
float area;
area= b*a;
return area;}
```

```
float perimetro(float ba,float al){
float p;
p=2*(ba*al);
return p;
}
```

Função

Escreva uma função que receba dois parâmetros: o valor do produto e o percentual de desconto. A função deverá calcular o novo valor do produto com o desconto.

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

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

```
float desconto(float,float);
```

```
int main (){
```

```
float v,d;
```

```
printf("Digite o valor do produto \n");
```

```
scanf("%f",&v);
```

```
printf("Digite o valor do desconto \n");
```

```
scanf("%f",&d);
```

```
printf("Valor com desconto: R$ %2.2f\n", desconto(v,d));
```

```
system("pause");
```

```
return 0;
```

```
}
```

```
float desconto(float valor, float desc){
```

```
    return valor-valor*desc/100;
```

```
}
```

Função

Escreva uma função para calcular o fatorial de um número. Faça também a função principal (main) que leia um número digitado pelo usuário e, utilizando a função, exiba o valor do fatorial calculado.

Lembrando:

$$5! = 5 * 4 * 3 * 2 * 1 = 120$$

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

```
int fatorial(int);
```

```
int main (){
```

```
int num;
```

```
printf("Digite um número\n");
```

```
scanf("%d",&num);
```

```
printf("O fatorial do número: %d\n", fatorial(num));
```

```
system("pause");
```

```
return 0;}
```

```
int fatorial(int n){
```

```
int fat=1,i;
```

```
for(i=n;i>0;i--){
```

```
    fat=fat*i; // fat*=i;
```

```
}
```

```
return fat;
```

```
}
```

Função

Escreva uma função que recebe um número inteiro n como parâmetro e retorne quantos números pares existem entre 1 e n . Escreva a função `main()` para testar.


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

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

```
int par(int);
```

```
int main (){
```

```
int num;
```

```
printf("Digite um número\n");
```

```
scanf("%d",&num);
```

```
printf("Números pares: %d\n", par(num));
```

```
system("pause");
```

```
return 0;
```

```
}
```

```
int par(int n){
```

```
int cont=0,i;
```

```
for(i=1;i<=n;i++){
```

```
    if(i%2==0){
```

```
        cont++;
```

```
    }
```

```
}
```

```
return cont;
```

```
}
```

OBRIGADA!