
Algoritmos

Estruturas de repetição

- ESTRUTURAS DE REPETIÇÃO

- Se uma ação se repete em um algoritmo, em vez de escrevê-la várias vezes, em certos casos podemos resumir anotando uma vez só e solicitando que ela se repita, usando uma das **estruturas de repetição**.
- Podemos pedir que uma ação (ou um conjunto de ações) seja executada um número *definido* ou *indefinido* de vezes, ou *enquanto* um estado permanecer ou *até que* um estado seja atingido.
- As principais estruturas de repetição são:
 - ENQUANTO...FACA
 - REPITA...ATE
 - PARA...ATE...(passo ?) FACA

Formas de escrita (depende bibliografia)

comando	expressão	número de execuções	condição de parada
ENQUANTO-FAÇA	início	?	expressão falsa
REPITA-ATÉ	fim	No mínimo 1	expressão verdadeira
PARA-FAÇA	não tem	ABS (valor-final - valor-inicial + passo)	variável = valor-final

Observa-se que todo comando ENQUANTO-FAÇA pode ser convertido para REPITA-ATÉ e vice-versa. E todo comando PARA-FAÇA pode ser convertido para ENQUANTO-FAÇA, mas nem todo comando ENQUANTO-FAÇA pode ser convertido para PARA-FAÇA.

Exemplo1-Visualg

- algoritmo "exemplo1"
- // Seção de Declarações
- var
- a, b, cont: inteiro
- inicio
- // Seção de Comandos
- escreva ("Informe a: ")
- leia(a)
- escreva ("Informe b: ")
- leia(b)
- para cont de a ate b passo 1 faca
- se cont mod 2 <> 0 entao
- escreval ("Numero impar = ", cont)
- fimse
- fimpara
- fimalgoritmo

exemplo2-Visualg

- algoritmo "exemplo2"
- // Seção de Declarações
- var
- idade, maiorid, menorid, conthabit, contF300, contF, contM: inteiro
- sexo: caracter
- salario, somasal, mediasal: real
-
- inicio
- // Seção de Comandos
- somasal<- 0
- conthabit<- 0
- contF<- 0
- contF300<- 0
- contM <- 0
- escreva ("Informe idade ou digite 0(zero) para sair: ")
- leia (idade)
-
- maiorid <- idade
- menorid <- idade
-
- enquanto idade <> 0 faca

exemplo2-Visualg

- repita
- escreva ("Informe sexo (F(eminino) | M(asculino) : ")
- leia (sexo)
- ate ((sexo = "F") ou (sexo = "M"))
- escreva ("Informe salario: ")
- leia (salario)
- //preparando resposta do item a
- somasal<- somasal+salario
- //preparando resposta do item b
- se maiorid < idade entao
- maiorid <- idade
- fimse
- se menorid > idade entao
- menorid <- idade
- fimse
- //preparando resposta do item c
- se sexo = "F" entao
- contF<- contF + 1
- se salario <= 300.00 entao
- contF300<- contF300 + 1
- fimse
- fimse
- //preparando resposta do item b
- se sexo = "M" entao
- contM <- contM + 1
- fimse

exemplo2-Visualg

- `conthabit<- conthabit + 1`
-
- `escreva ("Informe idade ou digite 0(zero) para sair: ")`
- `leia (idade)`
- `fimenquanto`
-
- `//respostas finais`
- `//item a`
- `mediasal<- somasal / conthabit`
- `escreval ("A media dos salarios = ", mediasal)`
-
- `//item b`
- `escreval ("Maior idade = ", maiorid)`
- `escreval ("Menor idade = ", menorid)`
-
- `//item c`
- `escreval ("Percentual de mulheres com salario ate 300,00 = ", contF300*100/contF , "% das mulheres")`
-
- `//item d`
- `escreval ("Quantidade de homens = ", contM)`
- `fimalgoritmo`

exemplo3-Visualg

- algoritmo "Exemplo3"
- **var**
- gerado, num, tentativas: inteiro
- **inicio**
- // Seção de Comandos
- tentativas<- 0
- aleatorio on
- leia (gerado)
- //limpatela
- escreval("****-----Tente acertar o numero aleatorio gerado-----****")
- aleatorio off
- repita
- escreval("Digite um numero [0 - 100] : ")
- leia(num)
- tentativas<- tentativas + 1
- se num = gerado entao
- escreval ("Parabens voce acertou!! em ", tentativas, " tentativas")
- senao
- se num < gerado entao
- escreval ("O numero digitado eh menor que o numero gerado. ")
- fimse
- se num > gerado entao
- escreval ("O numero digitado eh maior que o numero gerado. ")
- fimse
- fimse
- ate num = gerado
- **fimalgoritmo**

– Com este tipo de instrução podemos fazer um *contador*.
Veja como seria uma contagem até 10:

- `CONTADOR = 0`

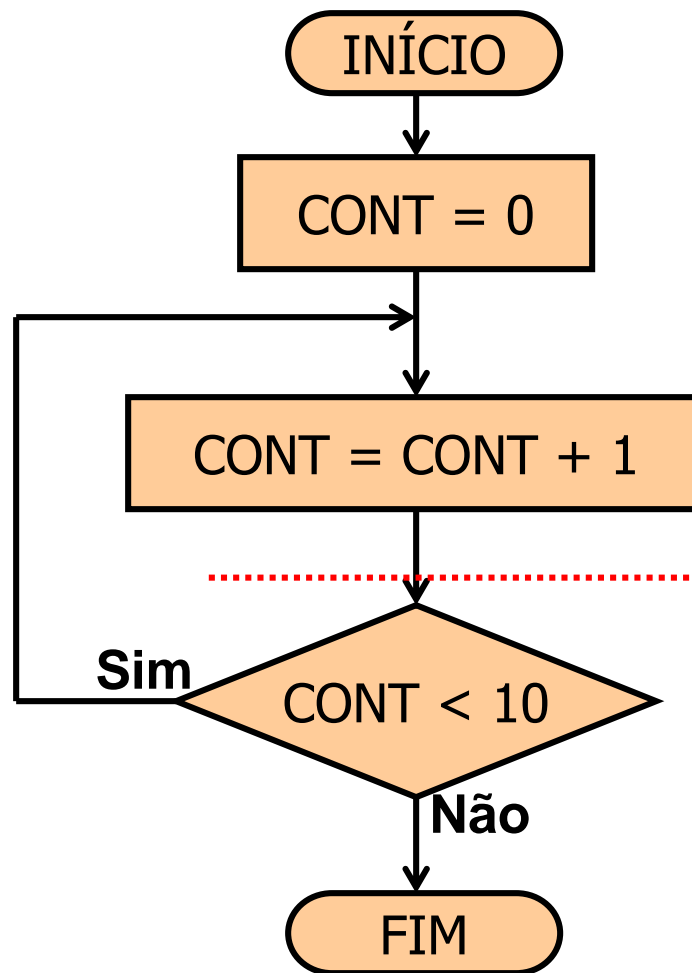
- **Repetir**

- `CONTADOR = CONTADOR + 1`

- enquanto** `CONTADOR < 10`



Isto será repetido
10 vezes.



Neste ponto do algoritmo podemos incluir qualquer conjunto de instruções que quisermos repetir 10 vezes.

Enquanto...Repetir

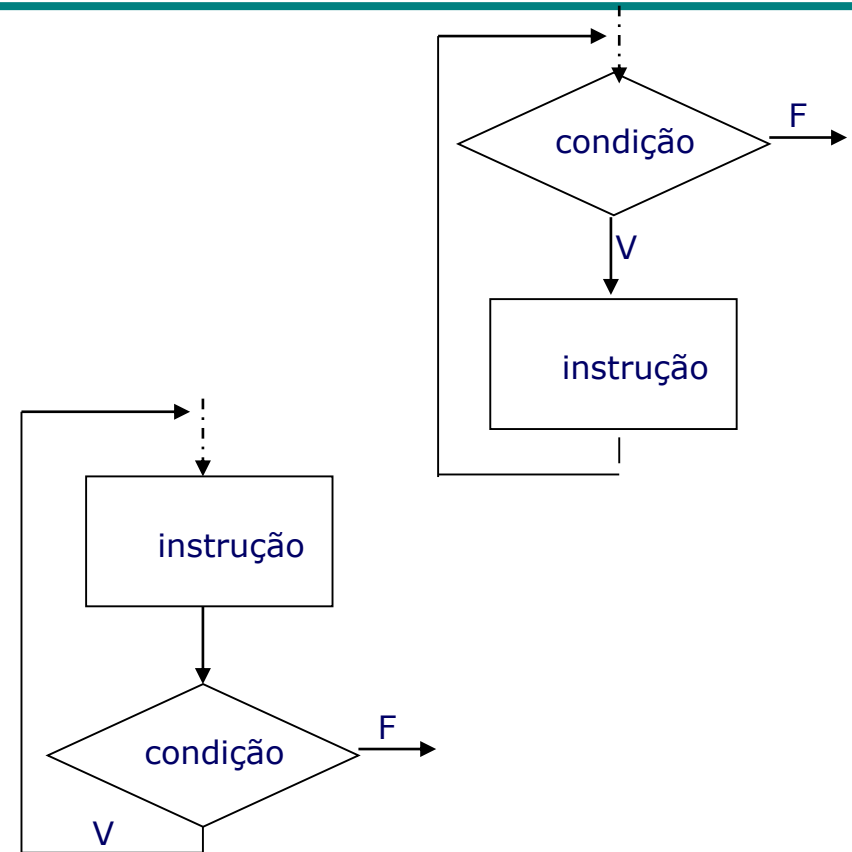
Enquanto (condição) repetir
<instruções>

Repetir...Enquanto

Repetir <instruções>
enquanto (condição)

Para...ate...repetir

Para <variavel> = <inicio> até <fim> repetir
<instruções>



Exemplo Enquanto

- **Ler 50 números fornecidos pelo usuário e calcular e exibir a média.**

Pseudocódigo:

Real: soma, num, media

Inteiro: cont

soma = 0

cont = 0

Enquanto cont < 50 repetir{

 ler num

 soma = soma + num

 cont = cont + 1

}

Media = soma / cont

Mostrar media

Exemplo Repita

Ler 50 números fornecidos pelo usuário e calcular e exibir a média.

Pseudocódigo:

Real: soma, num, media

Inteiro: cont

soma = 0

cont = 0

Repita{

 ler num

 soma = soma + num

 cont = cont + 1

}Enquanto cont <=50

Media = soma / cont

Mostrar media

Exemplo Para

- **Ler 50 números fornecidos pelo usuário e calcular e exibir a média.**

Pseudocódigo:

Real: soma, num, media

Inteiro: cont

soma = 0

Para cont =1 até 50 repetir {

ler num

soma = soma + num

}

Media = soma / cont

Mostrar media

- Até agora para vários valores informados pelo usuário líamos cada valor de forma separada
- Por exemplo, no algoritmo para o cálculo da média de quatro números, líamos 4 vezes, 4 valores para dentro de 4 variáveis.

Mas também poderíamos:

- ler um valor para 1 variável e repetir isso 4 vezes, adicionando cada valor lido ao total em uma outra variável, a cada repetição.
- Após as 4 repetições, a soma dos 4 números estaria acumulada na outra variável, bastando uma instrução para dividi-la por 4 e assim obter a média.

Média de notas de alunos em uma turma

real: n1, n2, n3, n4, soma,
media

soma=0

ler n1

ler n2

ler n3

ler n4

soma = n1+n2+n3+n4

media = soma/4

exibir media

real: n, soma, media

inteiro: i

soma=0

i=0

Repita{

ler n

soma = soma + n

i = i+1

}enquanto i<=4

media = soma/i

exibir media

PARA...ATÉ...REPETIR

– Formato:

Para <variável> = <valor inicial> **até** <valor final>
repetir <ações>

- Significado: A <variável> é inicializada com <valor inicial>. Após cada execução das <ações>, soma-se 1 à <variável> e repete-se as <ações>, continuando assim até que a <variável> atinja o <valor final>.
- Esta estrutura de repetição cria um *contador automático*, que nós não precisamos mandar incrementar.
- Ao usar esta estrutura já está subentendido que a <variável> inicia com <valor inicial> e é incrementada a cada ciclo (podendo-se inclusive aproveitar seu valor dentro do ciclo), e que as <ações> serão repetidas até que a <variável> tenha o <valor final>.

- Exemplo da estrutura PARA...ATÉ...REPETIR:
“Mostrar os quadrados dos inteiros de 3 a 11.”
 - Pseudocódigo:
Para CONT = 3 **até** 11 **repetir**
 Mostrar (CONT ** 2) } → Isto será repetido 9 vezes.
 - Usamos esta estrutura quando sabemos quantas vezes temos de repetir certas ações, mesmo que o número de vezes só seja conhecido *durante a execução*. Por exemplo:
“Perguntar ao usuário de quantos valores ele quer calcular a média. Ler os números e calcular a média.”



E se eu quisesse calcular a média de N números?

- Para esse problema construímos um algoritmo que será genérico, ou seja, que poderá ser usado para calcular a média de quantos números se quiser!
- Pseudocódigo:

Exibir “De quantos valores você quer calcular a média?”

Ler QUANT (aqui se descobre quantas repetições)

SOMA = 0

Para CONT = 1 até QUANT repetir

 Ler N (aqui é lido cada número, um em cada ciclo)

 SOMA = SOMA + N

MEDIA = SOMA / QUANT


Mostrar MEDIA



ENQUANTO ... REPETIR ...

– Formato:

Enquanto <operação lógica> **repetir** <ações>

- Significado: A <operação lógica> é testada. Se for verdadeira, então executar <ações> e em seguida testar novamente a op. lógica. Este ciclo prossegue até que em algum teste a op. lógica resulte em falso.
- Nesta estrutura temos novamente, assim como nas estruturas de decisão, uma *operação lógica* determinando se devemos *continuar* a repetir (resultado V) ou *parar* de repetir (resultado F) as ações. 
- Devemos garantir que o *dado testado* na op. lógica tenha seu *valor modificado* por alguma das ações repetidas, senão nunca teremos um resultado F no teste e a repetição permanecerá num ciclo infinito (loop)!

- Exemplo da estrutura ENQUANTO...REPETIR:
Pseudocódigo:

MAIOR = 0 (o maior por enquanto é o menor valor)

N = 1 (só para o 1.o teste funcionar...)

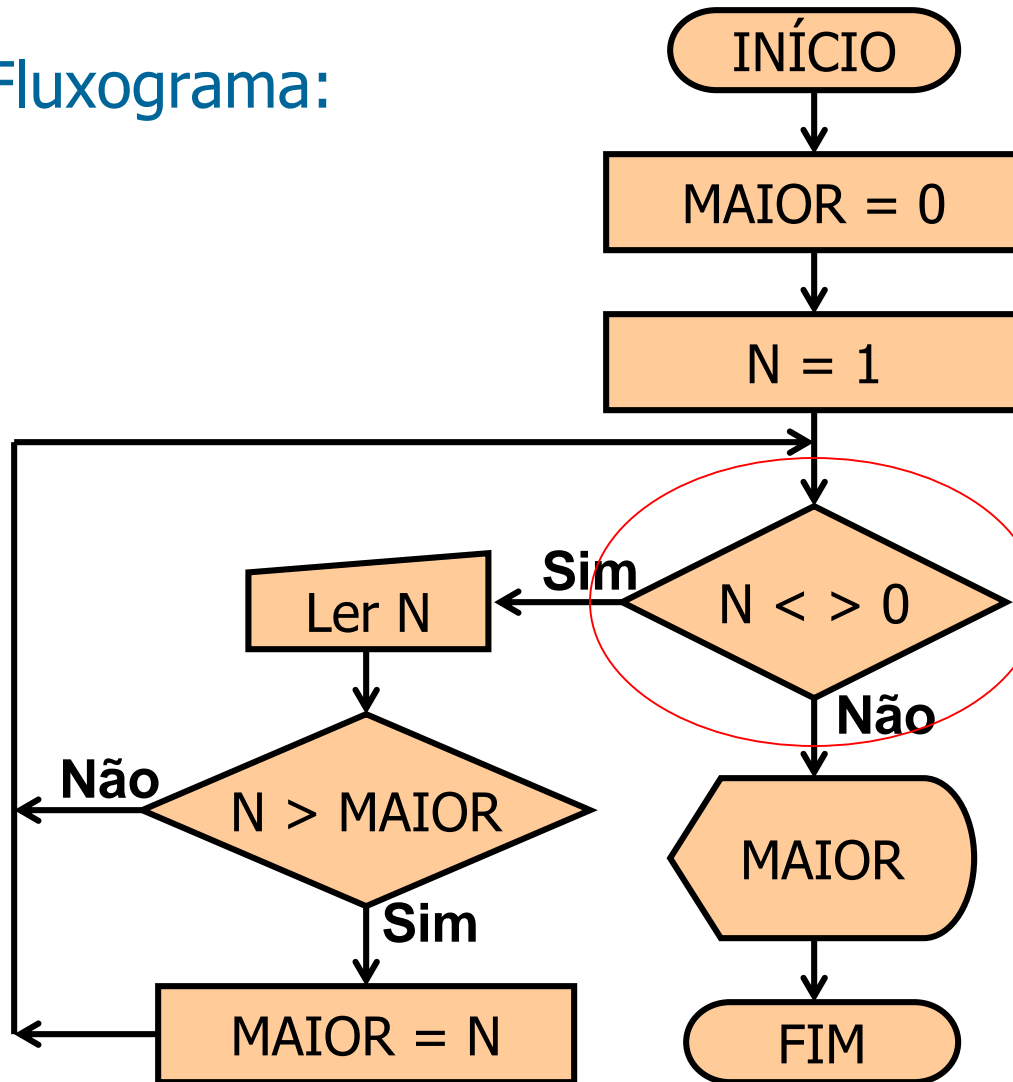
Enquanto (N <> 0) **repetir**

 Ler N (aqui o valor de N muda, é a entrada do usuário)

 Se (N > MAIOR) então MAIOR = N

Mostrar MAIOR (isto só executa qdo. o ciclo parar)

– Fluxograma:



Na estrutura ENQUANTO...REPETIR, veja que o teste é feito no *início* do ciclo.

Na Linguagem C

Estrutura de repetição

- Comando **for**

```
for (var=valor inicial; condição; incremento)
    comando;
```

Exemplo:

```
for (cont=3; cont<=11; cont++)
    cout<<cont<<endl;
```

```
for (var=valor inicial; condição; incremento)
{
    comando1;
    comando2;
    comando3;
}
```


Exercícios

Escrever um algoritmo que lê 5 valores, e conta quantos destes valores são negativos, escrevendo esta informação.

```
#include <iostream.h>
main(){
int numero, cont, neg=0;
for (cont=0; cont<5; cont++){
    cout<<"\nDigite um numero inteiro: ";
    cin>>numero;
    if (numero<0)
        neg++;
}
cout<<"\nO numero de valores negativos eh "<<neg;
system("pause");
}
```

Estrutura de repetição

- Comando **while**

```
while (condição)
    comando;
```

```
while (condição) {
    comando1;
    comando2;
    comando3;
}
```

Exemplo:

```
MAIOR=0;
N=1;
while (N != 0) {
    cin>>N;
    if (N > MAIOR) MAIOR = N;
}
cout<<"O numero maior eh "<< MAIOR;
system("pause");
```

Pseudo-código:

MAIOR = 0

N = 1

Enquanto (N <> 0) **repetir**

 Ler N

 Se (N > MAIOR) então MAIOR = N

 Mostrar MAIOR

Outra solução:

Escrever um programa que lê 5 valores, e conta quantos destes valores são negativos, escrevendo esta informação.

```
#include <iostream.h>
main(){
int numero, cont=0, neg=0;
while (cont<5){
    cout<<"\nDigite um numero inteiro: ";
    cin>>numero;
    if (numero<0)
        neg++;
    cont++;
}
cout<<"\nO numero de valores negativos eh "<<neg;
system("pause");
}
```

Estrutura de repetição

- Comando **do...while**

```
do {  
    comando  
} while (condição);
```

```
do {  
    comando1;  
    comando2;  
    comando3;  
} while (condição);
```

Exemplo:

```
cont=0;  
do {  
    cont = cont + 1;  
    cout<<cont<<endl;  
} while (cont < 10);
```

Em pseudo-código:

CONTADOR = 0

Repetir

CONTADOR = CONTADOR + 1

exibir CONTADOR

enquanto CONTADOR < 10

Outra solução:

Escrever um programa que lê 5 valores, e conta quantos destes valores são negativos, escrevendo esta informação.

```
#include <iostream.h>
main(){
int numero, cont=0, neg=0;
do{
    cout<<"\nDigite um numero inteiro: ";
    cin>>numero;
    if (numero<0)
        neg++;
    cont++;
}while (cont<5);
cout<<"\nO numero de valores negativos eh "<<neg;
system("pause");
}
```

Outros exemplos

1. Construir um algoritmo/programa C que calcule a média aritmética de vários valores inteiros positivos, lidos externamente. O final da leitura acontecerá quando for lido um valor negativo.

```
#include <iostream.h>
main(){
int numero, cont=0, soma=0;
float media;
cout<<"\nDigite um valor inteiro positivo. Digite um negativo para encerrar a execucao: ";
cin>>numero;
while (numero>=0)
{
soma=soma+numero;
cont++;
cout<<"\nDigite um valor inteiro positivo. Digite um negativo para encerrar a execucao: ";
cin>>numero;
}
if (cont>0){
media=soma/cont;
cin<<"\nA media eh "<<media;
}
else
cin<<"\nNenhum valor foi digitado";
system("pause");
}
```

Outra solução:

```
#include <iostream.h>
main(){
int numero, cont=0, soma=0;
float media;
cout<<"\nDigite um valor inteiro positivo. Digite um negativo para encerrar a execucao:";
cin>>numero;
if (numero >=0){
    do
    {
        soma=soma+numero;
        cont++;
        cout<<"\nDigite um valor inteiro positivo. Digite um negativo para encerrar a
            execucao: ";
        cin>>numero;
    }while(numero>=0);
    if (cont>0) {
        media=soma/cont;
        cout<<"\nA media eh "<<media;
    }
} //if
system("pause");
}
```


2. Escreva um algoritmo/programa C que calcule a média aritmética das 3 notas dos alunos de uma classe. O algoritmo deverá ler, além das notas, o código do aluno e deverá ser encerrado quando o código for igual a zero.

```
#include <iostream.h>
main(){
int codigo;
float media, nota, soma=0;
cout<<"\nDigite o codigo do aluno. Digite zero para encerrar a execucao: ";
cin>>codigo;
while (codigo!=0){
    soma=0;
    for (int cont=0; cont<3; cont++){
        cout>>"\nDigite nota: ";
        cin<<nota;
        soma=soma+nota;
    }//for
    media=soma/3;
    cout>>"\nA media do aluno "<<codigo<<" eh "<<media;
    cout<<"\nDigite o codigo do aluno.Digite zero para encerrar a execucao:";
    cin>>codigo;
} //while
system("pause");
}
```

Outra solução

```
#include <iostream.h>
main(){
int codigo;
float media, nota, soma=0;
cout<<"\nDigite o codigo do aluno. Digite zero para encerrar a execucao: ";
cin>>codigo;
do {
    soma=0;
    for (int cont=0; cont<3; cont++){
        cout<<"\nDigite nota: ";
        cout<<nota;
        soma=soma+nota;
    }//for
    media=soma/3;
    cout<<"\nA media do aluno "<<codigo<<" eh "<<media;
    cout<<"\nDigite o codigo do aluno. Digite zero para encerrar a execucao: ";
    cin>>codigo;
}while (codigo!=0);
system("pause");
}
```

3. Escreva um algoritmo/programa C que calcule a média dos números digitados pelo usuário, se eles forem pares. Termine a leitura se o usuário digitar zero (0).

```
#include <iostream.h>
main(){
int num, soma=0, cont=0;
float media;
cout<<"\nDigite um numero inteiro. Digite zero para encerrar a execucao: ";
cin>>num;
do {
    if (num%2==0){
        cont++;
        soma=soma+num;}
    cout<<"\nDigite um numero inteiro. Digite zero para encerrar a execucao: ";
    cin>>num;
}while (num!=0);
media=(float)soma/cont;
cout<<"\nA media eh "<<media;
system("pause");
}
```

Outra solução

```
#include <iostream.h>
main(){
int num, soma=0, cont=0;
float media;
cout<<"\nDigite um numero inteiro. Digite zero para encerrar a execucao: ";
cin>>num;
while (num!=0)
{
    if (num%2==0){
        soma=soma+num;
        cont++;
    }
    cout<<"\nDigite um numero inteiro. Digite zero para encerrar a execucao: ";
    cin>>num;
}
if (cont>0){
    media=soma/cont;
    cout<<"\nA media eh "<<media;
}
else
    cout<<"\nNenhum valor foi digitado";
system("pause");
}
```

Mais um exemplo...

4. Faça um programa C que realize a venda de produtos a clientes.

- Cada cliente pode comprar vários produtos.
- Sobre cada produto comprado por cada cliente, leia a quantidade de unidades compradas e o preço unitário.
- Encerre a entrada de produtos para um cliente quando o código do produto lido for -1.
- Mostre, para cada cliente, o valor total da sua compra.
- Encerre a entrada de clientes quando o código do cliente lido for zero.

```
#include <iostream.h>
main(){
int codProd, codCli, qtdade;
float precoUnit, valorTotal=0;
cout<<"\nDigite o codigo do cliente. Digite zero para encerrar a execucao: ";
cin>>codCli;
while (codCli!=0){
    valorTotal=0;
    cout<<"\n\tDigite o codigo do produto. Digite -1 para encerrar a execucao: ";
    cin>>codProd;
    while (codProd!=-1){
        cout<<"\n\tDigite a qtdade comprada pelo cliente "<<codCli<<" do produto "<<codProd;
        cin>>qtdade;
        cout<<"\n\tDigite o valor unitario do produto "<<codProd;
        cin>>precoUnit;
        valorTotal=valorTotal+qtdade*precoUnit;
        cout>>"\n\n\tDigite o codigo do produto. Digite -1 para encerrar a execucao: ";
        cin>>codProd;
    }
    cout<<"\nO cliente "<<codCli<<" deve pagar R$ "<<valorTotal;
    cout<<"\n\nDigite o codigo do cliente. Digite zero para encerrar a execucao: ";
    cin>>codCli;
} //while
system("pause");
}
```