

---

# Estruturas de repetição **do-while** e **for**

Prof. Karl Apaza Agüero

---

# Controle de repetição

---

## ► Estrutura de repetição: **do-while**

```
do
{
    <bloco de instruções>
}
while(condição);
```

```
while(condição)
{
    <bloco de instruções>;
}
```

A diferença entre o while e o do while é que o segundo já realiza o <bloco de instruções> pelo menos uma vez.

---

# Controle de repetição

---

## ► Exemplo 2: contar de 1 até 100

```
#include <stdio.h>
int main(){
    int cont = 1;
    do{
        printf("%d\n", cont);
        cont++;
    }while(cont <= 100);
    return 0;
}
```

# Problema

---

## ► Menu

Crie um menu que apresente as opções mostradas a seguir, e as execute a partir de um número ingressado pelo usuário:

MENU

0: Sair

1: Dar “Oi!” e sair

2: Repetir o menu

Também repetir o menu se outro número for ingressado.

---

```
#include <iostream>
using namespace std;
int main(){

    int opcao;
    do{
        cout<<"MENU"<<endl;
        cout<<"0: Sair"<<endl;
        cout<<"1: Dar Oi! e sair"<<endl;
        cout<<"2: Repetir o menu"<<endl;

        cin>>opcao;
        if(opcao==1){
            cout<<"Oi!"<<endl;
            opcao=0;
        }
    }while(opcao!=0);

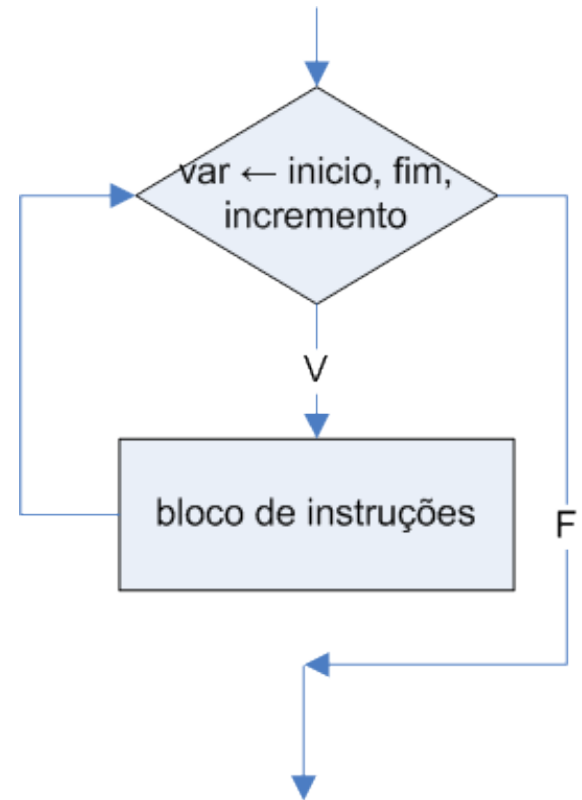
    return 0;
}
```

# Controle de repetição

---

- Estrutura de repetição: **for**  
Para...Faça

```
for (inicialização da variável;  
condição; atualização da variável)  
{  
    <bloco de instruções>  
}
```



# Controle de repetição

---

## ► Exemplo 3: contar de 1 até 100

### Usando **for**

```
#include <stdio.h>
int main(){
    for(int cont=1; cont<=100; cont++){
        printf("%d\n",cont);
    }
    return 0;
}
```

### Usando **while**

```
#include <stdio.h>
int main(){
    int cont = 1;
    while(cont <= 100){
        printf("%d\n",cont);
        cont++;
    }
    return 0;
}
```

---

## PROBLEMA

Sempre que Aparecido e seus colegas saíam para se alimentar, Aparecido pedia que algum de seus colegas fizesse a conta para decidir quanto era a conta total. Aparecido pediu sua ajuda para saber quando seus colegas acertaram o valor da conta total.

## ENTRADA

A entrada é composta por três linhas. A primeira linha do caso de teste possui um número inteiro **N** ( $1 \leq N \leq 10$ ) que indica quantas pessoas saíram juntas. A segunda linha do caso de teste possui **N** inteiros **C<sub>i</sub>** ( $1 \leq C_i \leq 100$ ) representando o valor da conta de cada uma das pessoas. A terceira linha possui um número inteiro **T** ( $1 \leq T \leq 5000$ ) representando a soma que o colega conseguiu realizar.

## SAÍDA

A saída é composta por uma única linha com a palavra **Acertou** quando o colega acertou a soma ou **Errou** quando ele errou a conta.

## EXEMPLOS

Entrada	Saída
3 1 5 8 14	Acertou
5 1 2 3 4 5 16	Errou



```
#include <iostream>
using namespace std;
int main(){
    int N;
    cin>>N;

    int i, conta_pessoa, conta_total=0;
    for(i=0; i<N; i++){
        cin>>conta_pessoa;
        conta_total+=conta_pessoa;
    }

    int T;
    cin>>T;
    if(T==conta_total) cout<<"Acertou"<<endl;
    else cout<<"Errou"<<endl;

    return 0;
}
```

# Problema

---

## ► Série de Fibonacci

Fazer um programa para gerar a série de Fibonacci:

1 1 2 3 5 8 13 21 34 55 89

até o  $n$ -ésimo elemento informado pelo usuário ( $n > 0$ ).

---

```
#include <iostream>
using namespace std;
int main(){
    int n;
    cin>>n;
    int a=1, b=1;
    if(n>=1) cout<<a<<" ";
    if(n>=2) cout<<b<<" ";

    for(int i=2; i<n; i++){
        int c=a+b;
        cout<<c<<" ";
        a=b;
        b=c;
    }
    cout<<endl;

    return 0;
}
```