

Algoritmos e Estrutura de Dados I

Estruturas de Repetição em C++

EXERCÍCIO TABUADA

- ▶ Faça um algoritmo onde o usuário digita um número e o algoritmo retorne (imprima na tela) a tabuada do número digitado. Considere a tabuada de 0 até 10.

ESTRUTURA DE REPETIÇÃO

- ▶ Utilizada quando um trecho ou até mesmo o algoritmo inteiro precisa ser repetido
- ▶ O número de repetições pode ser fixo ou estar associado a uma condição

ESTRUTURA DE REPETIÇÃO

- ▶ ESTRUTURA **PARA**

- ▶ ESTRUTURA **ENQUANTO**

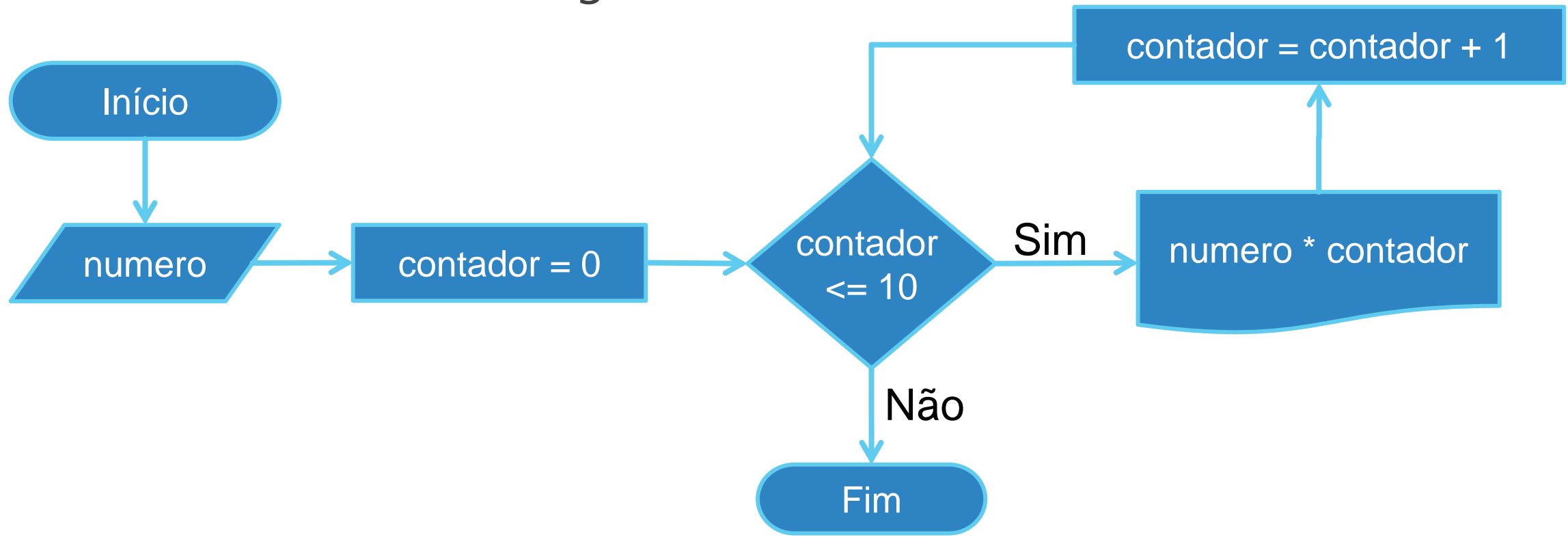
- ▶ ESTRUTURA **REPITA**

ESTRUTURA PARA

- ▶ Estrutura de repetição para número definido de repetições;
- ▶ É utilizada quando se sabe o número de vezes que um trecho do algoritmo precisa ser repetido

Estrutura “Para” em Fluxograma

- **Exemplo 1:** Faça um algoritmo onde o usuário deverá digitar um número e o algoritmo deverá retornar (imprimir na tela) a tabuada do número digitado. Considere a tabuada de 0 até 10.



ESTRUTURA PARA

PARA $i \leftarrow \text{valor_inicial}$ ATÉ valor_final FAÇA [PASSO n]

INÍCIO

comando1

comando2

.

.

FIM

*Comandos a
serem repetidos*

Quando houver apenas um
comando para repetir,
INÍCIO e **FIM** poderão ser
suprimidos

ESTRUTURA FOR

int i ;

for($i = \text{valor_inicial}$; $i \leq \text{valor_final}$; $i = i + n$)

{

comando1;

comando2;

.

.

}

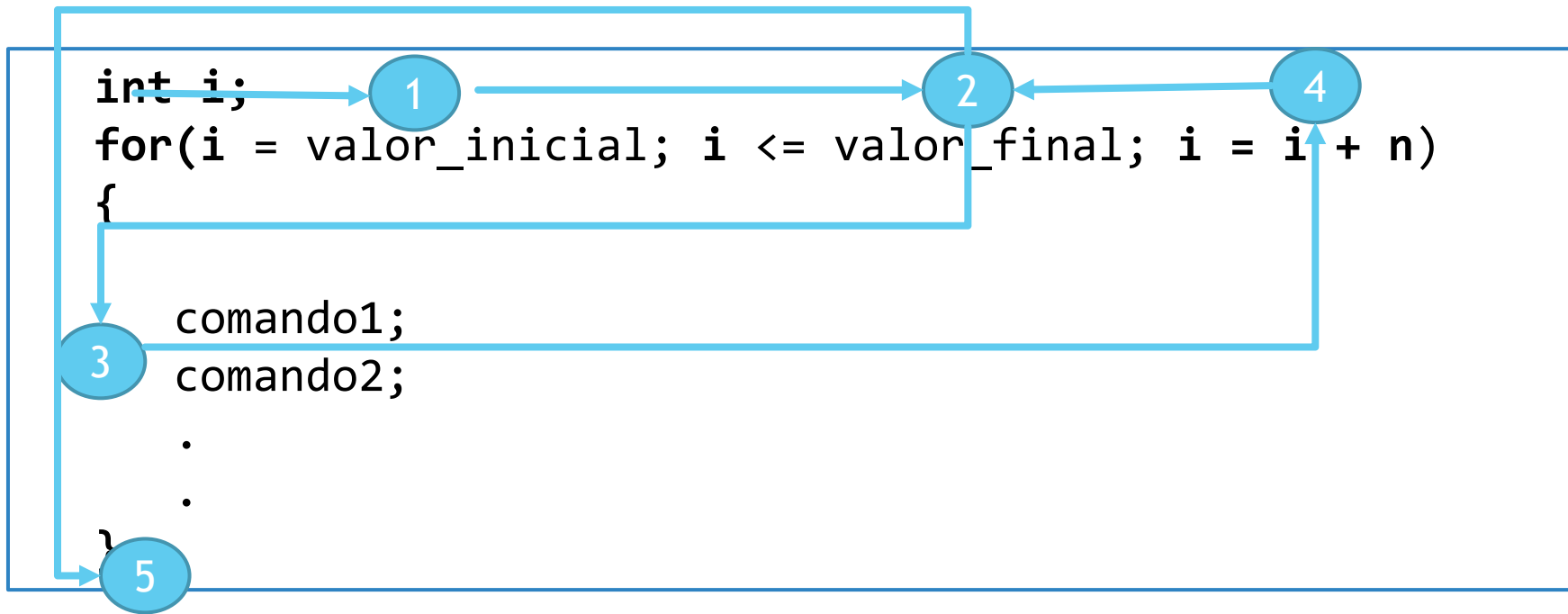
*Comandos a
serem repetidos*

ou
< valor_final
>= valor_final
> valor_final

ou
 $i - n$

Quando houver apenas um
comando para repetir, { e }
poderão ser suprimidos

ESTRUTURA FOR



ESTRUTURA FOR

EXEMPLOS

```
DECLARE i NUMÉRICO  
PARA i ← 1 ATÉ 10 FAÇA  
  ESCREVA i
```



```
int i;  
for(i = 1; i <= 10; i = i + 1)  
  cout << i;
```

```
DECLARE i NUMÉRICO  
PARA i ← 10 ATÉ 5 FAÇA  
  ESCREVA i
```



```
int i;  
for(i = 10; i >= 5; i = i - 1)  
  cout << i;
```

```
DECLARE i NUMÉRICO  
PARA i ← 1 ATÉ 9 FAÇA PASSO 2  
  ESCREVA i
```



```
int i;  
for(i = 1; i <= 9; i = i + 2)  
  cout << i;
```

```
DECLARE i NUMÉRICO  
PARA i ← 15 ATÉ 1 FAÇA PASSO -2  
  ESCREVA i
```



```
int i;  
for(i = 15; i >= 1; i = i - 2)  
  cout << i;
```

Repetições– Exemplo prático



ESTRUTURA FOR EXERCÍCIOS

- ▶ Crie um programa que receba 5 números e calcule a soma de seus valores. Utilize a estrutura de repetição PARA / FOR.

```
int i;  
for(i = valor_inicial; i <= valor_final; i = i + n)  
{  
    comando1;  
    comando2;  
    .  
    .  
}
```

ALGORITMO "SOMA"

DECLARE i, numero, soma NUMÉRICO

soma \leftarrow 0

PARA i \leftarrow 1 ATÉ 5 FAÇA

INÍCIO

 ESCREVA "DIGITE UM NÚMERO: "

 LEIA numero

 soma \leftarrow soma + numero

FIM

ESCREVA "Soma: ", soma

FIM_ALGORITMO

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int i, numero, soma;
```

```
    soma = 0;
```

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

```
    {
```

```
        cout << "DIGITE UM NUMERO:";
```

```
        cin >> numero;
```

```
        soma = soma + numero;
```

```
    }
```

```
    cout << "Soma: " << soma;
```

```
}
```

ESTRUTURA FOR EXERCÍCIOS

- ▶ Crie um programa que receba 5 números e mostre a quantidade de números maiores que 10. Utilize a estrutura de repetição PARA.

ALGORITMO "SOMA"

DECLARE i, numero, cont NUMÉRICO

cont ← 0

PARA i ← 1 ATÉ 5 FAÇA

INÍCIO

ESCREVA "DIGITE UM NÚMERO: "

LEIA numero

SE numero > 7

ENTÃO cont ← cont + 1

FIM

ESCREVA "Quantidade de números maiores
que 7: ", cont

FIM_ALGORITMO

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int i, numero, cont;
```

```
    cont = 0;
```

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

```
    {
```

```
        cout << "DIGITE UM NUMERO: ";
```

```
        cin >> numero;
```

```
        if(numero > 10)
```

```
            cont = cont + 1;
```

```
    }
```

```
    cout << "Quantidade de numeros maiores que  
7: " << cont;
```

```
    return 0;
```

```
}
```

ESTRUTURA FOR

EXERCÍCIOS

- ▶ Crie um programa que receba 2 notas de 6 alunos.
- ▶ Calcule e mostre:
 - ▶ A média aritmética das duas notas de cada aluno e a situação do aluno de acordo com a tabela abaixo;
 - ▶ O total de alunos aprovados, de recuperação e reprovados;
 - ▶ A média da classe;

Média	Situação
< 4	Reprovado
>= 4 e < 6	Recuperação
>=6	Aprovado

ESTRUTURA FOR

EXERCÍCIOS

ALGORITMO "Notas"

```
DECLARE i, nota1, nota2, media, total_media NUMÉRICO
total_aprovados, total_recuperacao, total_reprovados NUMÉRICO
total_media ← 0
total_aprovados ← 0
total_recuperacao ← 0
total_reprovados ← 0
PARA i ← 1 ATÉ 6 FAÇA
INÍCIO
    ESCREVA "DIGITE A NOTA 1: "
    LEIA nota1
    ESCREVA "DIGITE A NOTA 2: "
    LEIA nota2
    media ← (nota1 + nota2)/2
    ESCREVA "Média: ", media
    SE media < 4
    ENTÃO INÍCIO
        ESCREVA "Reprovado"
        total_reprovados ← total_reprovados + 1
    FIM
    SENÃO SE media >= 4 E media < 6
    ENTÃO INÍCIO
        ESCREVA "Recuperação"
        total_recuperacao ← total_recuperacao + 1
    FIM
    SENÃO INÍCIO
        ESCREVA "Aprovado"
        total_aprovados ← total_aprovados + 1
    FIM
    total_media ← total_media + media
FIM
ESCREVA "Total Reprovados: ", total_reprovados
ESCREVA "Total Recuperação: ", total_recuperacao
ESCREVA "Total Aprovados: ", total_aprovados
ESCREVA "Média da classe: ", total_media/6
FIM_ALGORITMO
```

```
#include <iostream>

using namespace std;

int main()
{
    int i, total_aprovados, total_recuperacao, total_reprovados;
    float nota1, nota2, media, total_media;
    total_media = 0;
    total_aprovados = 0;
    total_recuperacao = 0;
    total_reprovados = 0;
    for(i = 1; i <= 6; i++)
    {
        cout << "DIGITE A NOTA 1: ";
        cin >> nota1;
        cout << "DIGITE A NOTA 2: ";
        cin >> nota2;
        media = (nota1 + nota2)/2;
        cout << "Média: " << media;
        if(media < 4)
        {
            cout << "\nReprovado\n";
            total_reprovados = total_reprovados + 1;
        }
        else if(media >= 4 && media < 6)
        {
            cout << "\nRecuperação\n";
            total_recuperacao = total_recuperacao + 1;
        }
        else
        {
            cout << "\nAprovado\n";
            total_aprovados = total_aprovados + 1;
        }
        total_media = total_media + media;
    }
    cout << "\nTotal Reprovados:" << total_reprovados;
    cout << "\nTotal Recuperação: " << total_recuperacao;
    cout << "\nTotal Aprovados: " << total_aprovados;
    cout << "\nMédia da classe: " << total_media/6;

    return 0;
}
```


ESTRUTURA ENQUANTO

ESTRUTURA ENQUANTO

- ▶ Estrutura de repetição para número indefinido de repetições e teste no início;
- ▶ É utilizada quando não se sabe o número de vezes que um trecho do algoritmo precisa ser repetido, embora possa ser utilizada quando se conhece esse número

ESTRUTURA ENQUANTO

ENQUANTO condição **FAÇA**

INÍCIO

comando1

comando2

comando3

.

.

FIM

*Comandos a
serem repetidos*

Quando houver apenas
um comando para
repetir, **INÍCIO** e **FIM**
poderão ser suprimidos

Os comandos serão executados enquanto a condição for verdadeira

A repetição ocorrerá enquanto a condição for verdadeira

ESTRUTURA ENQUANTO

ENQUANTO condição FAÇA

INÍCIO

comando1

comando2

.

.

FIM

*Comandos a
serem repetidos*

Quando houver apenas um comando para repetir, **INÍCIO** e **FIM** poderão ser suprimidos

ESTRUTURA WHILE

```
int i = 0;
```

```
while(i < 10)
```

```
{
```

```
    comando1;
```

```
    comando2;
```

```
    i++;
```

```
    .
```

```
}
```

*Comandos a
serem repetidos*

Quando houver apenas um comando para repetir, { e } poderão ser suprimidos

ESTRUTURA REPITA

ESTRUTURA REPITA

- ▶ Estrutura de repetição para número indefinido de repetições e teste no final.
- ▶ É utilizada quando não se sabe o número de vezes que um trecho do algoritmo precisa ser repetido, embora possa ser utilizada quando se conhece esse número.

ESTRUTURA REPITA

REPITA

INÍCIO

comando1

comando2

comando3

.

.

FIM

ATE condição

*Comandos a
serem repetidos*

Quando houver apenas um comando para repetir, **INÍCIO** e **FIM** poderão ser suprimidos

Os comandos serão executados enquanto a condição for verdadeira

A repetição ocorrerá enquanto a condição for verdadeira

Os comandos serão executados pelo menos uma vez

ESTRUTURA REPITA

EXEMPLOS

ALGORITMO “REPETIÇÃO”

DECLARE x, y NUMÉRICO

$x \leftarrow 1$

$y \leftarrow 5$

REPITA

INÍCIO

$x \leftarrow x + 2$

$y \leftarrow y + 1$

FIM

ATE $x > y$

FIM_ALGORITMO

ALGORITMO “REPETIÇÃO”

DECLARE i NUMÉRICO

$i \leftarrow 0$

REPITA

INÍCIO

ESCREVA i

$i \leftarrow i + 1$

FIM

ATE $i > 10$

FIM_ALGORITMO

ESTRUTURA REPITA

REPITA

INÍCIO

comando1

comando2

.

FIM

ATE condição

*Comandos a
serem repetidos*

Quando houver apenas um comando para repetir, **INÍCIO** e **FIM** poderão ser suprimidos

ESTRUTURA DO WHILE

```
int i = 0;
```

```
do
```

```
{
```

```
    comando1;
```

```
    comando2;
```

```
    i++;
```

```
    .
```

```
}while(i < 10);
```

*Comandos a
serem repetidos*

Quando houver apenas um comando para repetir, { e } poderão ser suprimidos

ESTRUTURA REPITA EXERCÍCIOS

- ▶ Crie um algoritmo que receba 5 números e calcule a soma de seus valores. Utilize a estrutura de repetição REPITA.

```
#include <iostream>
int main()
{
    int i, numero, soma;
    soma = 0;
    i = 0;
    do{
        cout << "DIGITE UM NÚMERO: ";
        cin >> numero;
        soma = soma + numero;
        i = i + 1;
    } while(i < 5);
    cout<< "Soma: "<< soma;
}
```

Exercícios

1. Crie um programa que receba vários números enquanto o usuário quiser e calcule a soma de seus valores.
2. Escreva um programa que leia um numero inteiro e calcule a soma de todos os divisores desse numero, com exceção dele próprio.

Ex: a soma dos divisores do numero 45 é $1 + 3 + 5 + 9 + 15 = 33$

Exercícios

3. Faça um programa que some os números ímpares contidos em um intervalo definido pelo usuário. O usuário define o valor inicial e o valor final deste intervalo e o programa deve somar todos os números ímpares contidos neste intervalo.
- Caso o usuário digite um intervalo inválido (começando por um valor maior que o valor final) deve ser escrito uma mensagem de erro na tela, "Intervalo de valores inválido" e o programa termina.

Exemplo de tela de saída:

Digite o valor inicial : 5

Digite o valor final: 10

Soma dos ímpares neste intervalo: 21

Exercícios

4. Crie uma calculadora no C++. Leia o valor a , o operador ("+", "-", "*", "/") , a variável b e ao final informe o resultado da operação.
5. Quincas tem 1.85 metro e cresce 1 centímetro por ano, enquanto Zé Lelé tem 1.40 metros e cresce 3 centímetros por ano. Escreva um programa que calcule e imprima quantos anos ser ao necessários para que Zé Lelé e seja maior que Quincas.

Resolução 1

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int numero,soma=0;
```

```
    cout << "Digite um numero:";
```

```
    cin >> numero;
```

```
    while(numero != 0)
```

```
{
```

```
    soma += numero;
```

```
    cout << "Digite um numero:";
```

```
    cin >> numero;
```

```
}
```

```
cout << "Soma = " << soma;
```

```
return 0;
```

```
}
```

Resolução 2

```
#include <iostream>
using namespace std;
int main()
{
    int numero,soma=0;
    cout << "Digite um numero:";
    cin >> numero;

    for(int i=1; i<numero; i++)
    {
        if(numero %i ==0)
            soma += i;
    }
    cout << "Soma dos divisores = " << soma;
    return 0;
}
```

Resolução 3

```
#include <iostream>
using namespace std;

int main()
{
    int valini, valfinal, soma =0;
    cout << "Valor inicial do intervalo ";
    cin >> valini;
    cout << "Valor final do intervalo ";
    cin >> valfinal;
    if(valini <= valfinal) {
        for(int i= valini; i <= valfinal; i++)
        {
            if( i % 2 == 1)
                soma += i;
        }
        cout<< "Soma dos ímpares = "<<
        soma;
    } else
    {
        cout << "Intervalo de valores
        invalido!!";
    }
    return 0;
}
```


Resolução 4

```
#include <iostream>
using namespace std;
int main()
{
    float a, b, resp;
    char op;
    cout << "Digite um numero:";
    cin >> a;
    cout << "Digite o operador:";
    cin >> op;
    cout << "Digite um numero:";
    cin >> b;
    switch((op))
    {
        case '+':
            resp = a+b;
            break;
        case '-':
```

```
            resp = a-b;
            break;
        case '*':
            resp = a*b;
            break;
        case '/':
            if(b==0)
                resp=0;
            else
                resp = a/b;
            break;
        default:
            cout << "Operador não contemplado";
            break;
    }
    cout<< resp;
    return 0;
}
```

Resolução 5

```
#include <iostream>
using namespace std;
int main()
{
    float altQ=1.85,altZ=1.4;
    int ano=0;
    do{
        altQ += 0.01;
        altZ += 0.03;
        ano++;
    }while(altQ > altZ);
    cout<< "São necessários " << ano << "anos";
    return 0;
}
```