

# VALIDAÇÃO DE LINHAS DE CÓDIGO

## Estruturas de repetição com variável de controle

Os trechos de algoritmos a seguir processam um conjunto de dados único para obter os resultados desejados.

1º Exemplo: ler 3 notas de um aluno, calcular e exibir a média aritmética dessas notas.

### // Código em Portugol

```
funcao inicio ()
{
    Real nota1, nota2, nota3, media
    escreva (" nota 1 = ")
    leia (nota1)
    escreva (" nota 2 = ")
    leia (nota2)
    escreva (" nota 3 = ")
    leia (nota3)
    media = (nota1+nota2+nota3)/3
    escreva (" media = ",media)
}
```

### //Codigo em C

```
int main ()
{
    float nota1, nota2,nota3,media;
    printf (" nota 1 = ");
    scanf("%f", &nota1);
    printf (" nota 2 = ");
    scanf("%f", &nota2);
    printf (" nota 3 = ");
    scanf("%f", &nota3);
    media=(nota1+nota2+nota3)/3;
    printf ("media= %f",media);
}
```

2º Exemplo: ler 3 notas de um aluno, calcular e mostrar a média aritmética e exibir se o aluno foi aprovado ou não (média igual ou acima de 7 aprova o aluno).

Observe que a única diferença desta solução e a do exemplo anterior são as **duas últimas linhas**:

#### **// Código em Portugol**

```
funcao inicio ()  
{  
  Real nota1, nota2, nota3, media  
  escreva (" nota 1 = ")  
  leia (nota1)  
  escreva (" nota 2 = ")  
  leia (nota2)  
  escreva (" nota 3 = ")  
  leia (nota3)  
  media = (nota1+nota2+nota3)/3  
  escreva (" media = ",media)  
  se (media>=7) escreva(" , Aluno Aprovado")  
  senao escreva(" , Aluno Reprovado")  
}
```

#### **// Código em C**

```
float nota1, nota2,nota3,media;  
printf (" nota 1 = ");  
scanf("%f", &nota1);  
printf (" nota 2 = ");  
scanf("%f", &nota2);  
printf (" nota 3 = ");  
scanf("%f", &nota3);  
media=(nota1+nota2+nota3)/3;  
printf ("media= %f",media);  
  if (media>=7)  
    printf (" , Aluno Aprovado");  
  else  
    printf (" , Aluno Reprovado");  
return 0;
```

Vimos dois exemplos que processam três notas de um único aluno. Mas e se quiséssemos calcular e mostrar a média e a situação de aprovação de 40 alunos de uma turma? Temos duas soluções de acordo com o que sabemos até o momento:

- Executar o programa 40 vezes.

- Usar 120 variáveis: 3 variáveis para as notas de cada um dos 40 alunos ( $3 \times 40 = 120$ ).

Você considera viável alguma dessas soluções? E se fossem 100, 300 ou 1.000 alunos?

A sequência de comandos para ler, calcular e mostrar a média e situação de um aluno é igual para 100 ou qualquer outra quantidade de alunos, concorda? Como devemos proceder então?

Os comandos de repetição nos permitem repetir, quantas vezes desejarmos, uma sequência ou bloco de comandos.

## Demonstração

Vamos ver um exemplo da utilização dos comandos de repetição.

**Objetivo:** ler 3 notas de 40 alunos, calcular e mostrar a média aritmética e a situação de aprovação de cada aluno (média igual ou acima de 7 aprova o aluno).

**Lógica:** para resolver esse problema, precisamos aprender as estruturas de repetição, também chamadas de comandos de repetição ou de iteração. Dessa forma, somos capazes de repetir uma sequência ou bloco de comandos na quantidade de vezes que precisamos.

Nesse exemplo, precisamos repetir 40 vezes a mesma sequência de comandos do programa do exemplo anterior, pois o procedimento de cálculo da média de um aluno é igual para cada um dos 40 alunos.

Uma das possíveis soluções para a repetição de uma sequência de comandos, em um programa, é o uso do comando classificado como estrutura de repetição com variável de controle, que possibilita a repetição de uma sequência ou bloco de comandos em um número fixo e conhecido de vezes.

É o mais indicado quando sabemos previamente, pelo enunciado do problema, o número de vezes que a repetição vai acontecer. Nesse exemplo, a premissa é que devemos processar notas de uma turma com 40 alunos. Sabemos, portanto, que vamos repetir o procedimento (sequência ou bloco de comandos) para calcular a média, mostrá-la e exibir a situação do aluno 40 vezes: uma vez para cada um dos 40 alunos que fazem parte da turma.

**Em Portugol (pseudocódigo), o comando é o PARA.**

**Na linguagem C, o comando é o FOR.**

A seguir, apresentamos a sintaxe geral do comando de repetição com variável de controle em Portugol e em C:

**// Código em Portugol**

**//Sintaxe geral do comando PARA**

**PARA (inicialização; condição; incremento\_decremento)**

**{**

**Bloco ou sequência de comandos a ser repetida**

**}**

**// comando após a repetição**

**// Código em Linguagem C**

**//Sintaxe geral do comando FOR**

**for (inicialização; condição; incremento\_decremento)**

**{**

**Bloco ou sequência de comandos a ser repetida**

**}**

**// comando após a repetição**

O comando é composto de 3 partes:

## Inicialização

Valor inicial da variável de controle. Essa ação é executada uma única vez ao iniciar o comando.

## Condição

Expressão relacional (retorna um valor verdadeiro ou falso) associada à variável de controle. A condição é avaliada antes da repetição: se for verdade, a repetição ocorre; se for falsa, a repetição não ocorre e o fluxo do programa vai para o comando após a repetição.

## Incremento\_decremento

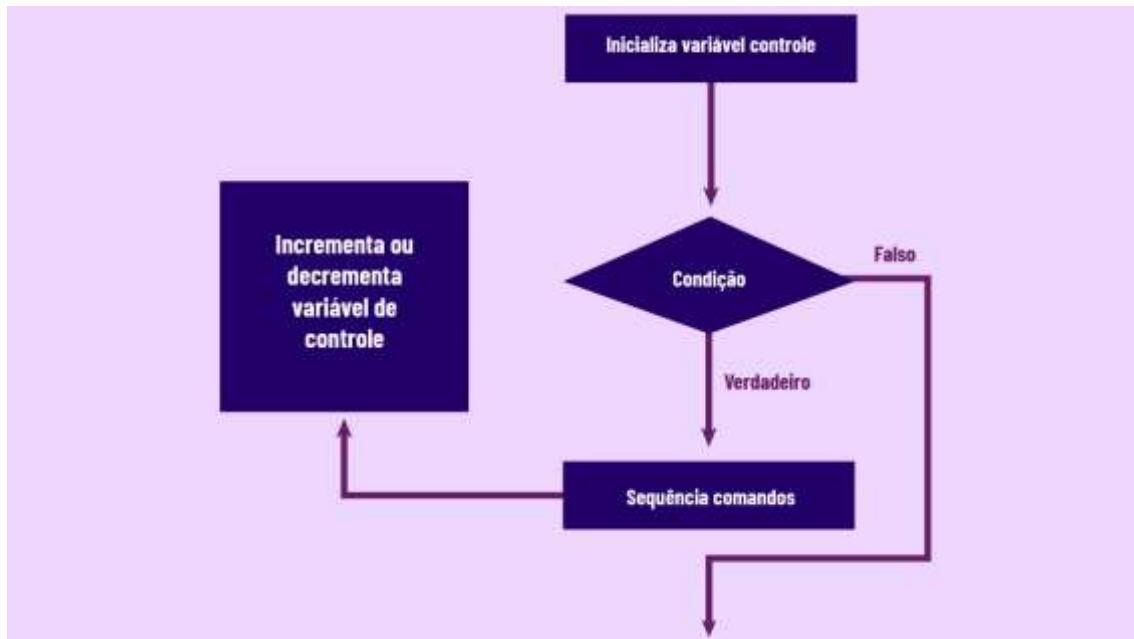
Aumento ou diminuição do valor da variável de controle ao fim da sequência de comandos a ser repetida. O incremento ou decremento pode ser de qualquer valor, conforme a solução desejada.

# Funcionamento do comando FOR

Essa estrutura de repetição usa uma variável que controla cada vez que o bloco ou a sequência de comandos será repetido. Chamamos de laço cada ciclo de repetição da sequência de comandos.

1. Essa variável de controle recebe o valor inicial, definido na inicialização.
2. O valor da variável de controle é comparado com a condição, que define o fim da repetição.
3. Se a condição for verdadeira:
  1. **Primeiro passo:** o bloco, ou sequência de comandos, a ser repetido é executado.
  2. **Segundo passo:** o valor da variável de controle é alterado (incrementado ou decrementado), conforme o problema, em incremento\_decremento.
  3. **Terceiro passo:** volta-se ao segundo passo.
4. Se a condição for falsa, a sequência de comandos a ser repetida é interrompida e o controle do código é passado ao comando após repetição.

Observe o fluxograma que ilustra o funcionamento da sintaxe geral do comando FOR:



Agora veja o uso dessa estrutura de repetição em Portugol e na linguagem C:

**// Código em Portugol**

```
PARA (cont=1;cont<=10;cont=cont+1)  
{  
Sequência de comandos a ser repetida  
}
```

**// Código em Linguagem C**

```
for (cont=1;cont<=10;cont=cont+1)  
{  
Sequência de comandos a ser repetida  
}
```

Em Portugol:

- Variável de controle: cont, do tipo inteiro.
- Inicialização: cont=1.
- Condição: cont<=10.
- Incremento\_decremento: cont=cont+1.

Na linguagem C:

- Variável de controle: cont, do tipo inteiro.
- Inicialização: cont=1.

- Condição:  $\text{cont} \leq 10$ .
- Incremento\_decremento:  $\text{cont} = \text{cont} + 1$ .

Veja algumas situações em que podemos aplicar esse comando.

Mostrar os 10 primeiros números inteiros e positivos em ordem crescente:

```
//Portugol
PARA (cont=1;cont<=10;cont=cont+1)
{
escreva( cont, "\n")
}

//C
for (cont=1;cont<=10;cont=cont+1)
{
    printf ("%d\n",cont);
}
```

Mostrar os 10 primeiros números inteiros e positivos em ordem decrescente:

```
Código Em Portugol
PARA (cont=10;cont>=1;cont=cont-1)
{
escreva( cont, "\n")
}

C
for (cont=10;cont>=1;cont=cont-1)
{
    printf ("%d\n",cont);
}
```

Mostrar os números pares entre 1 e 10 (inclusive):

```
Código Em Portugol
PARA (cont=2;cont<=10; cont=cont+2)
{
    escreva( cont, "\n")
}

C
for (cont=2;cont<=10; cont=cont+2)
{
    printf ("%d\n",cont);
}
```

```
}
```

Mostrar todas as dezenas entre 0 e 100 (inclusive), em ordem crescente:

**Código Em Portugol**

```
PARA (cont=10;cont<=100; cont=cont+10)
{
    escreva( cont, "\n")
}
```

**C**

```
for (cont=10;cont<=100;cont=cont+10)
{
    printf ("%d\n",cont);
}
```

Mostrar todas as centenas entre 0 e 1000 (inclusive), em ordem decrescente:

**Código Em Portugol**

```
PARA (cont=1000;cont>=100; cont=cont-100)
{
    escreva( cont, "\n")
}
```

**C**

```
FOR (cont=1000;cont>=100; cont=cont-100)
{
    printf ("%d\n",cont);
}
```

Mostrar a soma dos números inteiros e positivos entre 1 e 10:

**Código Em Portugol**

```
Soma=0
PARA (cont=1;cont<=10; cont=cont+1)
{
    Soma=Soma+cont
}
escreva (Soma, "\n")
```

**C**

```
Soma=0;
for (cont=1;cont<=10;cont=cont+1)
{
    soma=soma+cont;
}
```



```
printf ("A soma dos números entre 1 e 10 =%d\n",soma);
```

Mostrar a média aritmética dos números inteiros e positivos entre 1 e 10:

**Código Em Portugol**

```
Soma=0  
PARA (cont=1;cont<=10; cont++)  
{  
    Soma=Soma+cont  
}  
Média=Soma/10  
escreva (media, "\n")
```

```
C  
soma=0;  
for (cont=1;cont<=10;cont++)  
{  
    soma=soma+cont;  
}  
media=soma/10;  
printf ("A média dos números entre 1 e 10= %f\n",media);
```

## Colocando a teoria em prática

Para praticarmos, selecionamos alguns problemas para você resolver usando algoritmos em Portugol e na linguagem C.

### 1º problema

Desenvolva um programa que leia um número e o mostre 20 vezes.

### 2º problema

Desenvolva um programa que leia 15 números inteiros e positivos e mostre o maior deles.

**Lógica:** precisamos de 3 variáveis do tipo inteiro (int) para armazenar cada número a ser lido, o maior dos números e controlar a repetição.

1. Inicializar a variável maior com zero.

2. Repetir 20 vezes (comando de repetição PARA):
  1. Ler o número (comando de entrada de dados).
  2. Se o número for superior à variável maior, ela recebe o conteúdo do número lido.
3. Exibir o conteúdo da variável maior (comando de exibição de dados).

Veja a seguir a solução do problema:

#### Portugol

```
{
    inteiro num,cont,maior;
    maior=0
    para (cont=1;cont<=15;cont=cont+1)
    {
        escreva ("Digite um número: ")
        leia (num)
        se num> maior
        {
            maior=num
        }
        {
            escreva ("O maior dos números lidos = ",maior)
        }
    }
}
```

#### C

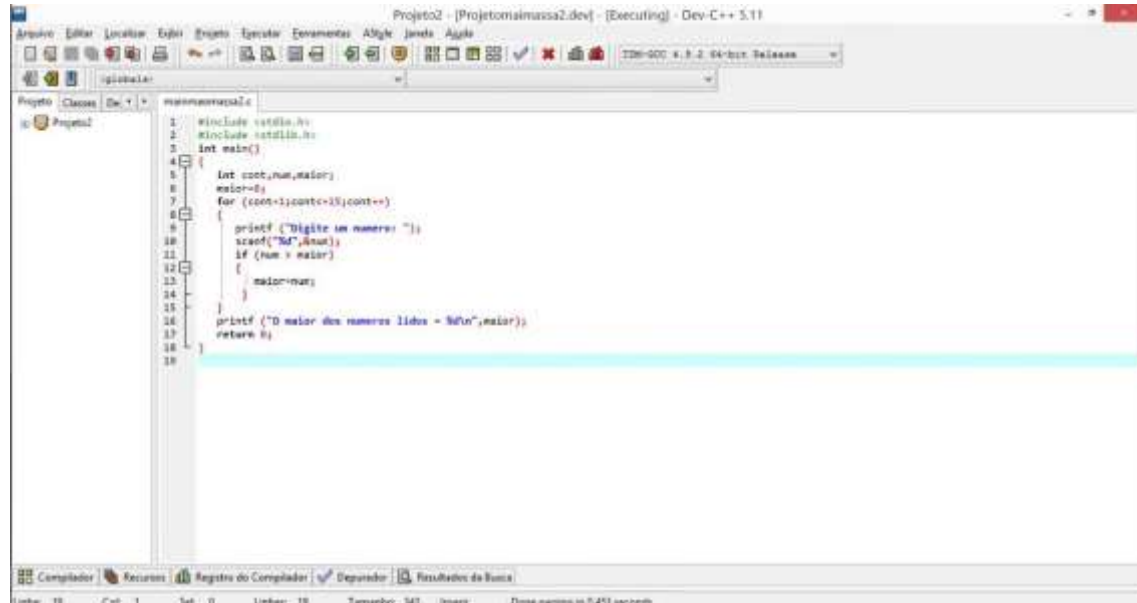
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int cont,num,maior;
    maior=0;
    for (cont=1;cont<=15;cont++)
    {
        printf ("Digite um número: ");
        scanf("%d",&num);
        if (num > maior)
        {
            maior=num;
        }
    }
}
```

```

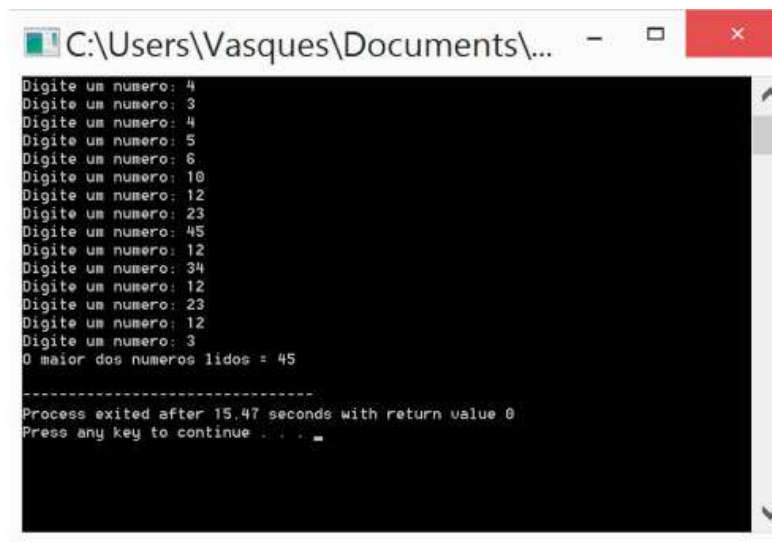
printf ("O maior dos números lidos = %d\n",maior);
return 0;
}

```

Para finalizar esse problema, veja o ambiente da ferramenta Dev-C++ com o código na linguagem C e a tela com o resultado da execução no ambiente Dev-C++.



**Ambiente da ferramenta Dev-C++ com o código na linguagem C**



**Resultado da execução no ambiente Dev-C++**

### 3º problema

Desenvolva um programa que leia o salário de 10 funcionários de uma empresa, calcule e mostre o maior salário e a média salarial da empresa.

**Lógica:** precisamos de 4 variáveis do tipo real (float), para armazenar cada salário a ser lido, o maior salário, a soma salarial — para calcular a média — e a média salarial, além de uma variável inteira (int) para controlar a repetição.

1. Inicializar com zero as variáveis: maior e soma.
2. Repetir 10 vezes (comando de repetição PARA):
  1. Ler o salário do funcionário (comando de entrada de dados).
  2. Se o salário for superior à variável maior, ela recebe o conteúdo do salário lido.
  3. Acumular a soma dos salários na variável soma.
3. Calcular a média salarial, dividindo a soma dos salários por 10 (total de funcionários).
4. Exibir o conteúdo das variáveis maior e média (comando de exibição de dados).

A seguir vemos a solução:

**Portugol**

programa

```
{
    funcao inicio()
    {
        inteiro cont
        real salario,media,soma,maior;
        maior=0 soma=0;
        para (cont=1;cont<=10;cont++)
        {
            escreva ("Digite o salário do funcionário: ");
            leia (salario);
            soma=soma+salario;
            se (salario > maior)
            {
                maior=salario;
            }
        }
    }
}
```

**C**

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

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

```
int main()
{
    int cont;

    float salario,media,soma,maior;

    maior=0; soma=0;

    for (cont=1;cont<=10;cont++)
    {
        printf ("Digite o salário do funcionário: \n");

        scanf("%f",&salario);

        soma=soma+salario;

        if (salario > maior)
        {
            maior=salario;
        }
    }

    media=soma/10;

    printf ("O maior salário da empresa e = %.2f \n",maior);

    printf ("A média salarial da empresa e = %.2f \n",media);

    return 0;
}
```

## 4º problema

Desenvolva um programa que leia 3 notas de 40 alunos, calcule e mostre a média aritmética e a situação de aprovação de cada um deles. Lembre-se de que apenas a média igual ou acima de 7 aprova o aluno.

**Lógica:** precisamos de 3 variáveis do tipo real (float) para armazenar as notas de cada aluno, uma variável real para armazenar a média das notas e uma variável inteira (int) para controlar a repetição.

1. Repetir 40 vezes (comando de repetição PARA):
  1. Ler nota1, nota2 e nota3 de cada aluno (comando de entrada de dados).
  2. Calcular a média do aluno:  $(\text{nota1} + \text{nota2} + \text{nota3}) / 3$ .
  3. Se a média do aluno for  $\geq 7$ , exibir aluno aprovado e sua média; senão, exibir aluno reprovado e sua média.

#### Portugol

programa

```
{
    funcao inicio()
    {
        real nota1,nota2,nota3,media
        inteiro contalunos
        para (contalunos=1;contalunos<=40;contalunos++)
        {
            escreva("Entre com a nota 1 do aluno: ")
            leia (nota1)
            escreva("Entre com a nota 2 do aluno: ")
            leia (nota2)
            escreva("Entre com a nota 3 do aluno: ")
            leia (nota3)
            media=(nota1+nota2+nota3)/3
        }
    }
}
```

#### C

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

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

```
int main()
```

```
{
```

```
    float nota1,nota2,nota3,media;
```

```
    int contalunos;
```

```

for (contalunos=1;contalunos<=6;contalunos++)
{
    printf("Entre com a nota 1 do aluno: \n");
    scanf("%f", &nota1);
    printf("Entre com a nota 2 do aluno: \n");
    scanf("%f", &nota2);
    printf("Entre com a nota 3 do aluno: \n");
    scanf("%f", &nota3);
    media=(nota1+nota2+nota3)/3;
    if (media>=7){
        printf("Aluno APROVADO com média : %.2f \n",media);
    } else {
        printf("Aluno REPROVADO com média : %.2f \n",media);
    }
}
return 0;
}

```

## 5º problema

Desenvolva um programa que leia, inicialmente, a porcentagem de reajuste dos salários dos funcionários de uma empresa. Na sequência, deve ler o salário de cada um dos 50 funcionários, calcular e mostrar o novo salário reajustado, aplicando a porcentagem de ajuste sobre os respectivos salários atuais. Ao final, o maior salário reajustado da empresa deve ser apresentado na tela.

**Lógica:** precisamos de 4 variáveis do tipo real (float) para armazenar a porcentagem de reajuste, cada salário a ser lido, cada salário a ser reajustado e o maior salário reajustado, além de uma variável inteira (int) para controlar a repetição.

1. Ler a porcentagem de reajuste (comando de entrada de dados).
2. Inicializar com zero a variável *maiorsal*.
3. Repetir 40 vezes (comando de repetição PARA).
4. Ler o salário do funcionário (comando de entrada de dados).
5. Calcular o salário reajustado, aplicando a porcentagem de aumento lido.
6. Se *salarioreajustado > maiorsalario*  
*maiorsalario = salarioreajustado*.
7. Exibir o conteúdo da variável *maiorsalario*.

// Código em Portugol

programa

```
{
    funcao inicio()
    {
        real percreaj, salario, salarioreaj, maiorsal
        inteiro cont
        maiorsal=0
        escreva("Percentual de reajuste salarial: ")
        leia (percreaj);
        para (cont=1;cont<=50;cont++)
        {
            escreva("Informe o salário do funcionário: ");
            leia (salario)
            salarioreaj = salario+ (salario*percreaj/100)
```

**C**

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

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

```
int main()
```

```
{
    float percreaj,salario,salarioreaj,maiorsal;
    int cont;
    maiorsal=0;
    printf("Percentual de reajuste salarial: \n");
    scanf ("%f",&percreaj);
```



```

for (cont=1;cont<=8;cont++)
{
    printf("Informe o salário do funcionário:\n");
    scanf("%f",&salario);
    salarioreaj=salario+(salario*percreaj/100);
    printf("O salário reajustado e : %.2f \n\n",salarioreaj);
    if (salarioreaj>maiorsal)
        maiorsal=salarioreaj;
}
printf("O maior salário reajustado e : %.2f",maiorsal);
return 0;
}

```

## 6º problema

Desenvolva um programa que leia um número N e, em seguida, uma lista de N números inteiros. Esse programa também deve calcular e mostrar a soma dos números pares e dos números ímpares da lista.

**Lógica:** precisamos de 4 variáveis inteiras (int) para armazenar o número N, cada número da lista de N números, a soma dos pares e a soma dos ímpares, além da variável para controle da repetição da leitura e processamento dos N números.

1. Ler o número N (comando de entrada de dados).
2. Inicializar as variáveis contadoras: somapar e somaimpar.
3. Repetir N vezes:
  1. Ler o número da lista.
  2. Se o resto da divisão do número da lista por 2 = 0  
*Somapar = somapar + número da lista*  
 Senão *somaimpar = somaimpar + número da lista.*
4. Exibir o conteúdo das variáveis *somapar* e *somaimpar*.

**Portugol**  
programa

```

{
    funcao inicio()
    {
        inteiro cont, n, num, somapar, somaimpar
        somapar=0
    }
}

```

```

    somaimpar=0
    escreva("Digite a quantidade de números da lista: ");
    leia(n);
    para (cont=1; cont<=n; cont++)
    {
        escreva ("Digite um número: ");
        leia(num);
        se (num%2==0)
            somapar=somapar+num;
c
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int cont,n,num,somapar,somaimpar;
    somapar=0;
    somaimpar=0;
    printf ("Digite a quantidade de números da lista: \n");
    scanf("%d",&n);
    for (cont=1;cont<=n;cont++)
    {
        printf ("Digite um número: \n");
        scanf("%d",&num);
        if (num%2==0)
            somapar=somapar+num;
        else
            somaimpar=somaimpar+num;
    }
    printf ("A soma dos números pares = %d\n",somapar);
    printf ("A soma dos números ímpares = %d\n",somaimpar);return 0;
}

```

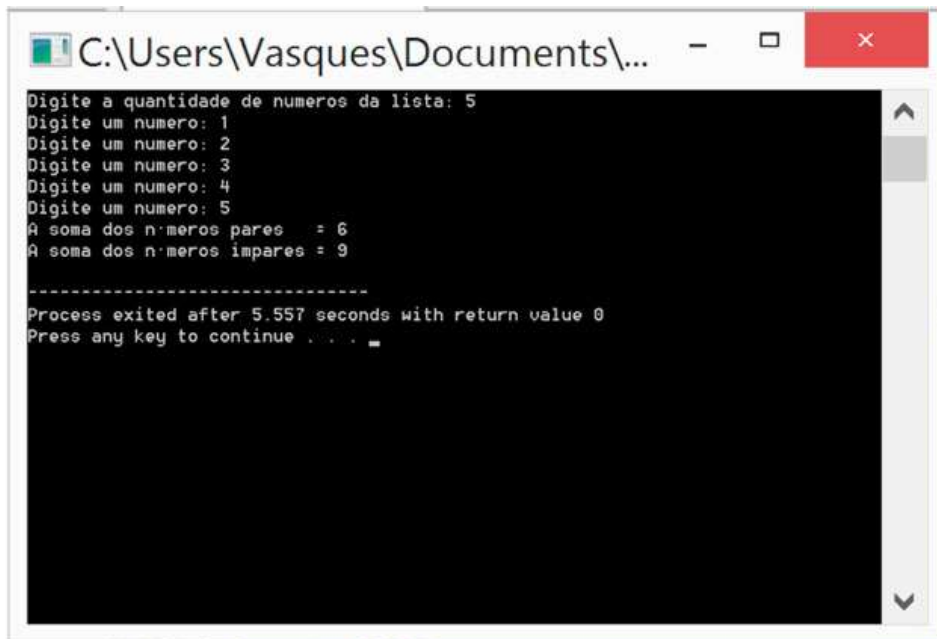
Para finalizar esse problema, veja o resultado da execução desse programa no Dev-C++:

O primeiro dado é a quantidade de números da lista: 5.

Na sequência, os 5 números: 1 2 3 4 5.

A soma dos pares:  $2 + 4 = 6$ .

A soma dos ímpares:  $1 + 3 + 5 = 9$ .



```
C:\Users\Vasques\Documents\... - [X]
Digite a quantidade de numeros da lista: 5
Digite um numero: 1
Digite um numero: 2
Digite um numero: 3
Digite um numero: 4
Digite um numero: 5
A soma dos n:meros pares = 6
A soma dos n:meros impares = 9

-----
Process exited after 5.557 seconds with return value 0
Press any key to continue . . .
```