**Sistema de Contagem de Números Positivos e Negativos em C**

**Descrição**

Este programa em linguagem C solicita ao usuário que digite 10 números inteiros e realiza a contagem de quantos são positivos e quantos são negativos. Ao final, o programa exibe:

* A quantidade de números positivos
* A quantidade de números negativos
* Uma mensagem indicando se houve mais números positivos, mais negativos ou se ocorreu um empate

O código demonstra o uso de estruturas de controle como while, if-else e operadores de comparação em linguagem C.

**Código Fonte**

c

#include <stdio.h>

int main() {

int n, positivos = 0, negativos = 0, i = 0;

printf("Digite 10 numeros inteiros:\n");

while (i < 10) {

printf("Numero %d: ", i + 1);

scanf("%d", &n);

if (n > 0) {

positivos++;

} else if (n < 0) {

negativos++;

}

i++;

}

printf("\nResultado:\n");

printf("Numeros positivos: %d\n", positivos);

printf("Numeros negativos: %d\n", negativos);

if (positivos > negativos) {

printf("Houve mais numeros positivos.\n");

} else if (negativos > positivos) {

printf("Houve mais numeros negativos.\n");

} else {

printf("Houve um empate entre numeros positivos e negativos.\n");

}

return 0;

}

**Funcionamento do Programa**

**1. Declaração de Variáveis**

c

int n, positivos = 0, negativos = 0, i = 0;

* n: Armazena cada número digitado pelo usuário
* positivos: Contador de números positivos (inicializado com 0)
* negativos: Contador de números negativos (inicializado com 0)
* i: Variável de controle do loop (inicializada com 0)

**2. Entrada de Dados**

c

while (i < 10) {

printf("Numero %d: ", i + 1);

scanf("%d", &n);

* O loop while executa 10 vezes (de 0 a 9)
* Para cada iteração, solicita um número ao usuário
* Usa scanf para ler o número inteiro digitado

**3. Classificação dos Números**

c

if (n > 0) {

positivos++;

} else if (n < 0) {

negativos++;

}

* Se o número for maior que 0, incrementa o contador de positivos
* Se o número for menor que 0, incrementa o contador de negativos
* Números zero não são contabilizados (não são positivos nem negativos)

**4. Exibição dos Resultados**

c

printf("\nResultado:\n");

printf("Numeros positivos: %d\n", positivos);

printf("Numeros negativos: %d\n", negativos);

* Exibe a quantidade de números positivos e negativos encontrados

**5. Análise Comparativa**

c

if (positivos > negativos) {

printf("Houve mais numeros positivos.\n");

} else if (negativos > positivos) {

printf("Houve mais numeros negativos.\n");

} else {

printf("Houve um empate entre numeros positivos e negativos.\n");

}

* Compara as quantidades de positivos e negativos
* Exibe mensagem apropriada indicando qual categoria prevaleceu ou se houve empate

**Exemplo de Execução**

text

Digite 10 numeros inteiros:

Numero 1: 5

Numero 2: -3

Numero 3: 0

Numero 4: 12

Numero 5: -7

Numero 6: 8

Numero 7: -1

Numero 8: 4

Numero 9: -2

Numero 10: 6

Resultado:

Numeros positivos: 5

Numeros negativos: 4

Houve mais numeros positivos.

**Características do Programa**

* **Simplicidade:** Código direto e fácil de entender
* **Eficiência:** Utiliza apenas um loop para processar todos os números
* **Completude:** Fornece análise completa com contagem e comparação
* **Robustez:** Considera o caso de empate entre positivos e negativos
* **Interface amigável:** Mensagens claras para o usuário durante a entrada e saída de dados

**Observações**

* O programa não considera o número zero (0) como positivo nem negativo
* A entrada de dados é sequencial e organizada
* A saída é formatada de maneira clara e informativa
* O código pode ser facilmente modificado para processar mais ou menos números alterando a condição do loop