

Programação 1

Operadores lógicos, seleção múltipla

Operadores lógicos

PROBLEMA:

Ler um número inteiro e imprimir a cor onde ele se encontra considerando a linha abaixo.



Solução 1

(sem o uso de operadores lógicos)

Leia N				
V	N < 10		F	
Escreva "Azul"	V	N<=50		F
	Escreva "Vermelho"		Escreva "Azul"	

Operadores lógicos

Operador	Operação	Prioridade
!	NÃO (not)	1 ^a
&&	E (and)	2 ^a
	OU (or)	3 ^a

A	B	A E B	A OU B	Não A
F	F	F	F	V
F	V	F	V	V
V	F	F	V	F
V	V	V	V	F

Solução 2 (utilizando o operador E)

10

50

Leia N		
V	N >= 10 E N <= 50	F
Escreva "Vermelho"		Escreva "Azul"

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

main()
{
    int n;

    printf("Informe um número inteiro:");
    scanf("%d", &n);

    if (n >= 10 && n <= 50)
        printf("Vermelho\n");
    else
        printf("Azul\n");
    system("pause");
}
```

Solução 3 (utilizando o operador OU)

10

50

Leia N	
V	N < 10 OU N > 50 F
Escreva "Azul"	Escreva "Vermelho"

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

main()
{
    int n;

    printf("Informe um número inteiro:");
    scanf("%d", &n);

    if (n < 10 || n > 50)
        printf("Azul\n");
    else
        printf("Vermelho\n");
    system("pause");
}
```

Seleção múltipla

PROBLEMA:

Escreva um algoritmo para ler 2 valores e o código da operação que deve ser executada sobre eles (1.Adição 2.Subtração 3.Multiplicação). Calcular e escrever o resultado da operação escolhida. Caso seja informado um código inválido escrever a mensagem "Operação inválida"

Solução 1 - Utilizando seleção aninhada

Leia n1,n2,op				
v		op = 1		F
R←n1+n2	v		op = 2	
	R ← n1-n2		F	
Escreva R	v		op = 3	
	R←n1 x n2		F	
	Escreva R		Escreva "Operação inválida"	

Seleção múltipla

Solução 2 - Utilizando a estrutura case (Seleção múltipla)

Leia n1,n2, op			
op			F
= 1	= 2	= 3	
$R \leftarrow n1 + n2$	$R \leftarrow n1 - n2$	$R \leftarrow n1 \times n2$	Escreva "Operação inválida"
Escreva R	Escreva R	Escreva R	

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

Comando switch

```
main() {
float n1,n2,r;
int op;

printf("Informe um valor:");
scanf("%f",&n1);
printf("Informe outro:");
scanf("%f",&n2);
printf("1.Adição 2.Subtração 3.Multiplicação:");
scanf("%d",&op);

switch(op) {
    case 1 : r = n1 + n2;
             printf("Resultado: %f\n",r);
             break;
    case 2 : r = n1 - n2;
             printf("Resultado: %f\n",r);
             break;
    case 3 : r = n1 * n2;
             printf("Resultado: %f\n",r);
             break;
    default: printf("Operação inválida\n");
}
system("pause");
}
```


Observações sobre o switch

- Testa apenas igualdade.
- A comparação é executada apenas com constantes.
- A variável utilizada no **switch** não deve ser do tipo ponto flutuante.

Forma geral

```
switch( variável)
{
    case const_A : comando1;
                  comando2;
                  ...
                  break;
    case const_B : comando3;
                  comando4;
                  ...
                  break;
    default      : comando5;
                  comando6;
                  ...
}
```

Optional

O problema do else “oscilante”

```
if (condição1)
  if (condição2)
    comando1;
  else
    comando2;
```

condição1		
V	F	
condição2		
V	F	
comando1	comando2	—

```
if (condição1)
  if (condição2)
    comando1;
else
  comando2;
```

condição1		
V	F	
condição2		
V	F	
comando1	—	comando2

Solução

```

if (condição1)
  if (condição2)
    comando1;
  else
    comando2;
  
```

condição1		
V	F	
condição2		
V	F	
comando1	comando2	—

```

if (condição1)
{
  if (condição2)
    comando1;
}
else
  comando2;
  
```

condição1		
V	F	
condição2		
V	F	
comando1	—	comando2