

## Exemplos de comandos de repetição em C while for do while

Caro aluno, qualquer dúvida que você tiver na implementação e teste destes programas, use o fórum para tirar suas dúvidas, ok?

Se você for recortar as soluções dos programas deste arquivo e colá-las no compilador DevC++, tome cuidado pois as aspas e o sinal de menos precisam ser substituídos, senão vai dar erro.

Outra dica: não use acentos nas palavras, pois não funcionam nos programas.

### 1) Comando while

---

Caro aluno, vejamos alguns exemplos de comandos `while` válidos:

```
contador=0;
while (contador < 3)
    ++contador;
```

---

```
numero=contapar = conta impar= 0;
while (numero <= 100)
{   if (numero % 2 == 0)
        ++contapar;
    else
        ++conta impar;
    numero += 3;
}
```

---

```
numero=contapar = conta impar= 0;
while (1)
{   if (numero % 2 == 0)
        ++contapar;
    else
        ++conta impar;
    numero += 3;
    if (numero >= 100)
        break;
}
```

---

Agora, vamos ver alguns exemplos de comandos `while` **inválidos** ou **errados**:

```
while contador <= 10
    ++contador;
```

*(Faltaram parênteses na condição)*

---

```
numero =1;
while (numero <= 100)
    if (numero % 2 == 0)
        ++contapar;
    else
        ++contaimpar;
    numero += 3;
```

*(Como não existem os delimitadores de início e fim { } do comando, somente o comando if faz parte do comando while e esse não altera a variável número que controla a condição, o comando while nunca vai parar.)*

---

```
numero = 1;
while (numero <= 10);
{    numero += 3;
}
```

*(Como foi colocado, após a condição, o comando while foi finalizado, a variável número não será alterada e o comando while nunca vai parar.)*

---

```
numero = 1;
while (1)
    numero += 3;
    if numero >= 10
        break;
```

*(Como não foram colocados { } delimitando os comandos dentro do while, que foi colocado em loop infinito de propósito, o comando break não será executado e o comando while nunca vai parar mesmo.)*

---

### Exemplo de um programa em C que usa o comando while

```
/*operacoes.c: Le vários pares de numeros inteiros n1,n2 e realiza
as 4 operacoes soma, subtracao, multiplicacao e divisao */
#include<stdio.h>
#include<conio.h>

int n1,n2,soma,subtr,mult;
float divisao;

int main()
{   while (1)
    {
        printf("\nDigite um numero inteiro ou -1 para finalizar: ");
        scanf("%i",&n1);

        if (n1 == -1)
            break;

        printf("\nDigte outro numero inteiro ");
        scanf("%i",&n2);

        soma=n1+n2;
        printf("\na soma de %i+%i=%i",n1,n2,soma);

        subtr=n1-n2;
        printf("\na subtracao de %i-%i",n1,n2,subtr);

        mult=n1*n2;
        printf("\na multiplicacao de %i+%i=%i",n1,n2,mult);

        if (n2!=0)
        {   divisao=n1/n2;
            printf("\na divisao de %i/%i=.3f",n1,n2,divisao);
        }
        else
        {   printf("\nnao existe divisao por zero");
        }
    }
    printf("\n\nfim do programa");
    getch();
    return 0;
}
```

### Exemplo de programa em C usando o comando while:

```
/* while.c - Solicita quatro numeros e os retorna na tela
   Ilustra o uso do comando while*/

#include <stdio.h>
#include <conio.h>
int vetor[4];
int indice=0, numero;

int main( )
{ printf("Le quatro numeros entre 1 e 100 e os mostra na tela \n");
  while (indice<4)
  {  numero=0;
    while( (numero<1) || (numero > 100) )
    {   printf("\nDigite um numero de 1 a 100: ");
        scanf("%i", &numero);
    }
    vetor[indice]=numero;
    indice++;
  }
  printf("\n\nOs numeros validos digitados foram:");
  for (indice=0; indice < 4; indice++)
    printf("\n[%i]= %i ", indice, vetor[indice]);

  printf("\n\nFim do programa");
  getch();
  return 0;
}
```

## 2) Comando for

---

A expressão *condição* pode ser constituída de expressões conectadas por operadores lógicos.

### Exemplos:

```
for(i=0; (i<10 && vetor[i]!=0); i++)
    printf("\n %i", vetor[i]);
```

```
achou= i= 0;
for( ; (i<10 && achou==0); i++)
{   printf("\n %i", vetor[i]);
    achou = (vetor[i] == -1);
}
```

```
}
```

A instrução **for** pode ser usada para se inicializar um vetor.

Exemplo:

```
for(i=0; i <10; vetor[i++]=0);
```

Usando vírgulas, podem-se criar expressões constituídas de duas subexpressões, de forma a se realizar duas tarefas. Elas serão avaliadas da esquerda para a direita e toda a expressão será avaliada como o valor da expressão da direita.

Exemplo:

```
for(i=0, j=10; i<10; i++, j--)  
    printf("\n%d, %d", i, j);
```

A instrução **for** pode ser aninhada, ou seja, executada dentro de outra instrução **for**.

Exemplo:

```
for(i=0; i<10; i++)  
{ for(j=0; j<10; j++)  
    printf("\n%i", matriz[i][j]);  
}
```

Vejamos agora alguns **exemplos de comandos for válidos:**

```
for (i=10; i>0; i--)  
    printf("\nContador decrescente= %i",i);
```

---

```
i= soma = 0;  
for ( ; i < 10; i++)  
{    printf("\nDigite o valor da nota do aluno %i= ",i);  
    scanf("%f",&nota);  
    soma = soma + nota;  
}
```

---

```
for (j=1; j<=1000; j++)  
{    if (j % 2 == 0)  
        printf("\n %i eh par ",j);  
    else printf("\n %i eh impar ",j);  
}
```

---

```
for (j=1; j<=100; j++)  
{    printf("\nDigite um nro negativo ou 0 para finalizar: ");
```

---

```
scanf("%i", &n);  
if (n==0)  
    break;  
else printf("\nValor absoluto de %i = %i", n, abs(n));  
}
```

---

### Exemplos de comandos for inválidos:

```
for (i=10; i<0; i--)  
    printf("\nContador decrescente= %i", i);
```

*(O comando for não será executado nenhuma vez, pois o valor inicial é menor que o valor final e i-- indica contagem decrescente.)*

---

```
for (i=10; i>0; i--);  
    printf("\nContador decrescente= %i", i);
```

*(O comando for não vai fazer nada, pois tem um; indevido i-- ), que faz com que o for finalize – printf será executado uma única vez.)*

---

```
for (j=1; j<=1000; j--)  
{  
    if (j % 2 == 0)  
        printf("\n %i eh par ", j);  
    else printf("\n %i eh impar ", j);  
}
```

*(O comando for não vai parar nunca, pois a condição nunca vai ser falsa; o certo seria usar j++ e não j--).*

### Exemplo de programa em C que usa o comando for:

```
/*Multiplos3.c: gera multiplos de 3 menores que 500 */  
  
#include <stdio.h>  
#include <conio.h>  
int num;  
int main()  
{  
    printf("\n Sequencia de numeros multiplos de 3 menores que 500");  
  
    for (num=3; num<500; )  
    {  
        printf("\n Multiplo de 3=%i ", num);  
        num = num + 3;  
    }
```

```
}

printf ("\n\n Fim do programa");
getch();
return 0;
}
```

### 3) Comando do while

**Exemplo de comando do while válido:**

```
i=0;
do
{
    i++;
    printf("\n%d", i);
} while(i<10);
```

**Exemplo de programa em C usando o comando do while:**

```
/* Codigos.c - Fornece o codigo ASCII de uma tecla pressionada */

#include <stdio.h> /* Por causa da funcao printf */
#include <conio.h> /* Por causa da funcao getche */

#define FIM 26 /* Codigo ASCII de ^Z */

int main()
{
    int c;
    do
    {
        c=getche();
        printf(" Tecla: %i\n", c);
    } while(c != FIM );

    printf("\nFim do programa");
    getch();
    return 0;
}
```

### Exemplo de programa em C usando o comando do while:

```
/*.....
   Menul.c - Operacionaliza um menu simples. Ilustra o uso do comando "do while "
   .....*/

#include <stdio.h>
#include <conio.h>
int main()
{ int opcao;

  puts("*****");
  puts(" Ilustra como mostrar um menu de opcoes");
  puts("*****");
  puts("Pressione uma tecla para comecar...");
  getch();

  for(;;)
  { do
    { puts("\nEscolha uma das opcoes:\n");
      puts("1) Sair do programa.");
      puts("2) Inserir dados.");
      puts("3) Ver as informacoes.");
      puts("4) Obter uma informacao.");
      printf("\nEscolha sua opcao: ");
      scanf("%i", &opcao);
    } while(opcao<1 || opcao>4);

    switch(opcao)
    { case 1:
      printf("\nFim do programa");
      getch();
      exit(0); // interrompe o for(;;) e sai do programa
      break;

      case 2:
      printf ("\nInsere dados....");
      getch();
      break;

      case 3:
      printf ("\nImprime dados...");
      getch();
      break;

      case 4:
      printf ("\nConsulta dados....");
      getch();
      break;

      default:
      puts("\nVoce nao escolheu uma opcao valida!");
      getch();
    }
  }
}
```