



1.1) Instrução For

A instrução **for** é uma instrução de controle de fluxo para a repetição de um bloco de comandos. A sintaxe da instrução **for** é:

```
for (expressão1; expressão2; expressão3) {  
    <comandos>  
}
```

A expressão1 define a instrução de inicialização da variável de controle (podendo ocorrer mais de uma variável ou nenhuma). A expressão2 define a condição de término, e a expressão3 contém as instruções de incremento. O bloco contendo os comandos a serem repetidos deve ser executado um número controlado de vezes, sendo sempre verificada a condição de término.

```
/*Programa para calcular a soma dos N primeiros números  
naturais (1+2+3+...+N)*/  
  
#include <stdio.h>  
  
int main(){  
    int num, soma, i;  
  
    printf("Digite um valor inteiro:");  
    scanf("%d", &num);  
    soma = 0;  
    for( i = 1; i <= num; i++ ){  
        s = s + i;  
    }  
    printf("A soma é %d\n", soma);  
}
```

A expressão1 define a instrução de inicialização da variável de controle (podendo ocorrer mais de uma variável ou nenhuma). A expressão2 define a condição de término, e a expressão3 contém as instruções de incremento. O bloco contendo os comandos a serem repetidos deve ser executado um número controlado de vezes, sendo sempre verificada a condição de término.

Exercícios em linguagem C:

1) Dadas as instruções *for* abaixo, explique cada uma e mostre o resultado a ser impresso.

- A) `for (x=0; ((x>3) && (x<9)); x++){
 printf("%d", x);
}`
- B) `for (x=0, y=4; (y<9); x++, y+=2){
 printf("x=%d, y=%d ->", x, y);
}`

- C) `for (x=0,y=4,z=4000;z; z/=10){
printf("X");
}`
- D) `for (i=0; i<10; i++){
printf("\n%d",i+1);
}`
- E) `for(i = 0; i < 10; printf("%d\n", i), i++);`
- F) `for (; ;){
printf("ola");
}`
- G) `for (i=0, j=1; i<10; i++){
printf("%d", j);
}`
- H) `j=13;
for (i=0, j=1; i<10; i++){
printf("%d", j);
}
printf("%d", j);`
- I) `j=13;
for (int i=0, j=1; i<10; i++){
printf("%d", j);
}
printf("%d", j);`
- J) `for(c = 'a'; c <= 'z'; c++){
printf("ASCII(%c) = %d\n",c,c);
}`
- K) `max = 5;
c = 0;
for(;c < max;){
c++;
printf("c = %d\n",c);
}`

2) Analise e teste o código abaixo. Explique o uso e a finalidade do comando *break*.

```
#include <stdio.h>

int main() {
    int i = 0, max=5;
    for(;;){
        i++;
        if(i > max)
            break;
        printf("i = %d\n",i);
    }
}
```

3) Analise e teste o código abaixo. Explique o uso e a finalidade dos comandos *break* e *continue*.

```
#include <stdio.h>

int main() {
    for ( int i = 1 ; i <= 20 ; i++ ) {
        if ( i == 2 ) continue;
        if ( i == 5 ) break;
        printf("i = %d\n",i);
    }
}
```

4) Sejam as escalas Celsius e Fahrenheit para a representação de valores de temperaturas. Faça um programa para imprimir duas tabelas:

a) Celsius para Fahrenheit: Para os valores em Celsius de -20° a $+30^{\circ}$, em intervalos de 5° , obter os valores equivalentes na escala Fahrenheit.

b) Fahrenheit para Celsius: Para os valores em Fahrenheit de -10° a $+100^{\circ}$, em intervalos de 5° , obter os valores equivalentes na escala Celsius.

$$F = 32 + \frac{212 - 32}{100} \times C$$
$$C = (F - 32) \times \frac{100}{212 - 32}$$