

## Tópico 8: comando iterativo *for...*

### Auto Avaliação 8.1 - Soluções

---

Utilizando o comando iterativo *for...*, desenvolva os seguintes programas.

1. Faça um programa C que leia 5 valores e calcula e imprime a sua média aritmética.

#### Programa C:

```
//inserir cabeçalho
#include <stdlib.h>
#include <stdio.h>
# define  NUMVAL 5
int main( )
{
    int i;
    float valor, somatorio, media;
    somatorio = 0;
    for (i = 1; i <= NUMVAL; i++)
    {
        printf("\n Valor %d: ", i);
        scanf("%f", &valor);
        somatorio = somatorio + valor;
    }
    media = somatorio / NUMVAL;
    printf("\n media = %8.2f \n", media);
    system("PAUSE");
    return 0;
}
```

#### Comentários:

*Na tentativa de primeira execução do for, o i é inicializado com 1. É verificado se ele é menor ou igual à NUMVAL (uma constante do programa). Se a condição se verificar, o bloco subordinado ao for é executado pela primeira vez. Ao final, o i é acrescido de 1 (escrever i++ é o mesmo que escrever i = i + 1). Novamente a condição é verificada. Se ela continuar válida, o bloco do for é executado de novo, e assim sucessivamente, até que o i seja igual à NUMVAL, quando então o for executa pela última vez, e a instrução após o mesmo passa a ser executada.*

2. Faça um programa C que, primeiramente, leia o total de valores inteiros a serem lidos (*n*) e, a seguir, leia os *n* valores e encontre o maior valor do conjunto de valores lidos.

### Programa C:

```
//inserir cabeçalho...
#include <stdlib.h>
#include <stdio.h>
#define VALORES 5
int main( )
{
    int numval, num, maior , i;
    printf("Numero de valores a processar: ");
    scanf("%d", &numval);
    for (i = 1; i <= numval ; i++)
    {
        printf("Valor (inteiro): ");
        scanf( "%d", &num );
        if (i == 1)
            maior = num;
        if (num > maior)
            maior = num;
    }
    printf("Maior valor = %d \n" , maior);
    system("PAUSE");
    return 0;
}
```

### Comentários:

*Para determinar-se o maior valor de um conjunto lido, adotou-se a seguinte estratégia:*

- *Criou-se uma variável, de nome maior;*
- *Ao ser lido o primeiro valor do conjunto a processar, inicializou-se maior com esse primeiro valor (if (i == 1) maior = num);*
- *Em qualquer execução do for, comparou-se o valor lido em num com o valor armazenado em maior. Se o valor do momento fosse maior do que o valor armazenado em maior, o valor de maior era atualizado: if (num > maior) maior = num;*

*Concluído o laço de leitura dos valores, o conteúdo de maior foi apresentado.*

3. Estenda o programa 2 para que o mesmo também forneça a média dos valores lidos.

### Programa C:

```
//inserir cabeçalho...
#include <stdlib.h>
#include <stdio.h>
#define VALORES 5
int main( )
{
    int numval, num, maior , i, somatorio;
    float media;
    somatorio = 0;
    printf("Numero de valores a processar: ");
    scanf("%d", &numval);
    for (i = 1; i <= numval ; i++)
    {
        printf("Valor (inteiro): ");
        scanf( "%d", &num );
        if (i == 1)
            maior = num;
        if (num > maior)
            maior = num;
        somatorio = somatorio + num;
    }
    media = (float)somatorio / numval;
    printf ("Media = %8.2f\n", media);
    printf("Maior valor = %d \n" , maior);
    system("PAUSE");
    return 0;
}
```

### Comentários:

*Para também calcular a média dos valores lidos, acrescentou-se uma variável somatório, que a cada novo valor lido resulta igual ao seu conteúdo anterior mais esse valor. Ao terminar a leitura de todos os valores, fora do laço **for**, calculou-se e apresentou-se a média.*

*A variável somatório necessita ser inicializada antes de ser usada, uma vez que desde a primeira vez que é utilizada, para acumular os valores lidos, seu valor anterior é considerado:*

*"somatorio = somatorio + 1".*

4. Faça um programa C que leia valores inteiros e conta e imprime quantos valores negativos foram lidos.

### Programa C:

```
//inserir cabeçalho ...
#include <stdlib.h>
#include <stdio.h>
#define NUMVAL 5
int main( )
{
    int valor, conta_negativos, i;
    conta_negativos = 0;
    for (i = 1; i<= NUMVAL; i++)
    {
        printf("Forneca um valor inteiro: ");
        scanf("%d", &valor);
        if (valor < 0)
            conta_negativos ++;
    }
    printf("\n Dos %d valores lidos, %d sao negativos \n",
          NUMVAL, conta_negativos);
    system("PAUSE");
    return 0;
}
```

### Comentários:

*Para a contagem dos valores negativos foi criada a variável inteira `conta_negativos`. Para contagens deve-se sempre usar variáveis inteiras. Inicialmente `conta_negativos` é inicializada com zeros.*

*No laço implementado com o **for**, a cada execução um novo valor é lido, sempre na mesma variável `valor`, e para cada valor lido verifica-se se ele é negativo.*

*Se o valor lido é negativo, ou seja, se a condição `if (valor < 0)` se verificar, `conta_negativos` é incrementado de 1 (`conta_negativos++`).*

*Terminado o **for**, é apresentada a mensagem dando conta de quantos valores foram lidos e quantos eram negativos.*

5. Faça um programa C que apresente na tela uma tabela de conversão de graus Fahrenheit para Celsius, de -148 °F a 212 °F. Use uma variação de 10° F.

### Programa C:

```
//inserir cabeçalho...
#include <stdlib.h>
#include<stdio.h>
int main( )
{
    float celsius;
    int fahr;
    printf("\n      Fahrenheit    Celsius\n");
    for (fahr = -148; fahr <= 212; fahr += 10)
    {
        celsius = (5.0/9.0) * (fahr - 32);
        printf("      %10d      %8.2f  \n", fahr, celsius);
    }
    system("PAUSE");
    return 0;
}
```

A variável fahr é incrementada de 10 a cada execução do *for*.

### Comentários:

*O primeiro comando será ativado e aparecerá na tela a linha que funcionará como um cabeçalho para os valores que serão apresentados em seguida:*

*Fahrenheit Celsius.*

*O **for** será ativado.*

*Na tentativa de primeira execução do **for**, fahr será inicializado com -148. Será verificado se o valor de fahr é maior ou igual a +212. Se a condição se verificar, o bloco subordinado ao **for** será executado pela primeira vez. Ao final de cada execução do **for**, fahr será incrementado de 10 (escrever "fahr += 10" é o mesmo que escrever "fahr = fahr + 10"). Novamente a condição será verificada. Se ela novamente continuar válida, o bloco do **for** será executado, e assim sucessivamente, até que fahr seja igual a +212, quando então o **for** executará pela última vez, e a instrução após o mesmo passará a ser executada.*

6. Faça um programa C que imprime os números pares entre 50 e 99.

### Programa C:

```
//inserir cabeçalho...
#include <stdlib.h>
#include <stdio.h>
int main( )
{
    int i;
    printf ("\nValores pares entre 50 e 99\n");
    for (i = 50; i <= 99; i= i + 2)
        printf("%d ", i);
    system("PAUSE");
    return 0;
}
```

A variável **i** sofre um acréscimo de 2 a cada execução do **for**.

### Comentários:

*O primeiro comando será ativado e aparecerá na tela: Valores pares entre 50 e 99.*

*O **for** será ativado.*

*Na tentativa de primeira execução do **for**, o **i** será inicializado com 50. Será verificado se ele é menor ou igual a 99. Se a condição se verificar, o bloco subordinado ao **for** será executado pela primeira vez. Ao final, o **i** será acrescido de 2 ( $i = i + 2$ ). Novamente a condição será verificada. Se ela continuar válida, o bloco do **for** será executado de novo, e assim sucessivamente, até que o **i** seja igual a 99, quando então o **for** executará pela última vez, e a instrução após o mesmo passará a ser executada.*

7. Faça um programa C que imprime os números ímpares entre 99 e 50.

### Programa C:

```
//inserir cabeçalho ...
#include <stdlib.h>
#include <stdio.h>
#define LINF 50
#define LISUP 99
int main( )
{
    int i;
    printf ("\nValores impares entre 99 e 50 \n");
    for (i= 99; i >= 50; i--)
        if (i % 2 != 0)
            printf("%d  ", i);
    system("PAUSE");
    return 0;
}
```

A variável *i* é decrementada de um a cada execução do *for*.

### Comentários:

*Resultado da execução do comando **for**: os ímpares entre 99 e 50 serão apresentados. O *i* será inicializado em 99 na primeira execução do **for** e a cada nova execução do laço será decrementado de 1 (escrever-se *i--* é o mesmo que escrever-se *i = i - 1*). A cada execução do **for** o valor de *i* será dividido por 2 e o resto da divisão comparado com zero. O operador % é um operador de divisão que retorna o resto da mesma. Se o resto for diferente de zero, o valor de *i* será apresentado.*

8. Faça um programa C que leia a idade e a nota sobre o filme “O segredo”, fornecidas por espectadores, e conta quantos espectadores maiores de 40 anos que atribuíram nota 10 ao filme.

### Programa C:

```
//inserir cabeçalho...
#include <stdlib.h>
#include <stdio.h>
#define ESPECTADORES 8
int main( )
{
    int nota, idade , i, cont_maisde40;
    cont_maisde40 = 0;
    for (i = 1; i <= ESPECTADORES ; i++)
    {
        printf("Idade: ");
        scanf( "%d", &idade );
        printf("Nota (entre 1 e 10): ");
        scanf( "%d", &nota );
        if ((idade > 40) && (nota == 10))
            cont_maisde40++;
    }
    Printf ("Espectadores com mais de 40 anos que atribuíram\n");
    Printf ("nota 10 ao filme ""O Segredo"" = %d\n", cont_maisde40);
    system("PAUSE");
    return 0;
}
```

### Comentários:

*Para a contagem dos espectadores com mais de 40 anos, que atribuíram nota 10 ao filme, foi criada a variável inteira cont\_maisde40. Inicialmente cont\_maisde40 foi inicializada com zeros.*

**Atenção:** para contagens deve-se sempre usar variáveis inteiras!

*Para cada espectador são lidas idade e nota. Se idade é > 40, se nota atribuída é 10, então cont\_maisde40 é incrementado (cont\_maisde40++ o que é o mesmo que cont\_maisde40 = cont\_maisde40 + 1).*

*Terminado o for, o valor de cont\_maisde40 é apresentado.*

***Obs.:*** No printf final, cada vez que se desejava que na tela, na mensagem apresentada, aparecessem aspas duplas ("), foi necessário usar-se duas aspas.

*\n em um printf provoca mudança de linha na tela.*