

Linguagem C
Introdução à programação em Linguagem C

Exemplos de scripts

Emanuel Camacho

2015

earc96@hotmail.com
a33878@ubi.pt

1 Programas

1.1 Calcular a média de n números reais inseridos.

```
#include <stdio.h>

main()
{
    int i,x,n;
    float x;
    float soma = 0;
    float average;

    do{
        printf("Insira n, com n > 1.\n");
        scanf("%d",&n);
    }while( n <= 1 );

    for ( i = 1 ; i <= n ; i++ )
    {
        scanf("%f",&x);
        soma = soma + x;
    }

    average = soma/n;

    printf("\nA média dos valores inseridos é : %0.3f",average);
}
```

- **Nota :**

- *%0.3f* significa que o número mostrado é real com três casas decimais.

1.2 Determinar o maior e o menor número de uma sequência de n números inteiros.

```
#include <stdio.h>

main()
{
    int i,n;
    int x;
    int menor, maior;

    do{
        printf("Insira n, com n > 1.\n");
        scanf("%d",&n);
    }while( n <= 1 );

    scanf("%d",&x);
    menor = x;
    maior = x;

    for ( i = 1 ; i <= n-1 ; i++ )
    {
        scanf("%d",&x);

        if ( x >= maior )
        {
            maior = x;
        }

        if ( x <= menor )
        {
            menor = x;
        }
    }
    printf("O menor número inserido foi %d e o maior %d.",menor, maior);
}
```

1.3 Calcular o fatorial de N se $N > 1$ e a sua tangente se $N < 1$.

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

main()
{
    int N;
    int i;
    float result = 1;

    printf("Insira um N.\n");
    scanf("%d",&N);

    if( N > 1 )
    {
        for ( i = 1 ; i <= N ; i++ )
        {
            result = result * i;
        }
        printf("O resultado é %0.0f",result);
    }

    if ( N < 1 )
    {
        result = sin(N)/cos(N);
        printf("O resultado é %0.3f",result);
    }
}
```

1.4 Calcular o valor de x^k sem usar a função predefinida.

OBS:

$$x^k = \begin{cases} x^k & , k > 0 \\ 1 & , k = 0 \\ \frac{1}{x^k} & , k < 0 \end{cases}$$

```
#include <stdio.h>
main()
{
    float x;
    int k;
    float result = 1;

    printf("Insira o valor de x e k.");
    scanf("%f %d",&x,&k);

    if ( k == 0 )
    {
        printf("O resultado é 1.");
    }
    else if ( k < 0 )
    {
        while ( k < 0 )
        {
            result = result * 1/x;
            k++;
        }
        printf("O resultado é %0.3f",result);
    }
    else
    {
        while ( k > 0 )
        {
            result = result * x;
            k--;
        }
        printf("O resultado é %0.3f",result);
    }
}
```

1.5 Calcular o valor da expressão com $N \geq 1$.

$$\text{soma} = \sum_{k=1}^N \begin{cases} \frac{1}{k \cdot (k+1)} & , k \text{ é par} \\ -\frac{1}{k \cdot (k+1)} & , k \text{ é ímpar} \end{cases}$$

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

```
main()
{
    int k,n;
    float soma = 0;

    do{
        printf("Insira N com N >= 1\n");
        scanf("%d",&n);
    }while( n < 1 );

    for ( k = 1 ; k <= n ; k++ )
    {
        if ( k%2 == 0 )
        {
            soma = soma + (1.0/(k*(k+1)));
        }
        else
        {
            soma = soma - (1.0/(k*(k+1)));
        }
    }
    printf("\nA soma é %0.4f",soma);
}
```

- **Nota :**

- Foi colocado 1.0 em vez de 1 nas linhas de código em que se está a calcular a soma, de modo a evitar problemas de processamento. Estes problemas devem-se ao estar a utilizar dois tipos de variáveis diferentes (*int* e *float*) ao mesmo tempo.

1.6 Inverter um vetor com n elementos.

```
#include <stdio.h>

main()
{
    int n;

    do{
        printf("Insira um N com N > 1\n");
        scanf("%d",&n);
    }while( n <= 1 );

    int v[n], u[n];
    int i;
    int j = n - 1;

    for ( i = 0 ; i < n ; i++ )
    {
        scanf("%d",&v[i]);
    }

    for ( i = 0 ; i < n ; i++ )
    {
        u[i] = v[j];
        j--;
    }
}
```

1.7 Determinar quantos múltiplos e divisores de 7 um vetor tem.

```
#include <stdio.h>

main()
{
    int n;

    do{
        printf("Insira um N com N > 1\n");
        scanf("%d",&n);
    }while( n <= 1 );

    int v[n];
    int i;
    int Multiplos7 = 0;
    int Divisores7 = 0;

    for ( i = 0 ; i < n ; i++ )
    {
        scanf("%d",&v[i]);
    }

    for ( i = 0 ; i < n ; i++ )
    {
        if ( v[i]%7 == 0 )
        {
            Multiplos7 = Multiplos7 + 1;
        }
        if ( 7%v[i] == 0 )
        {
            Divisores7 = Divisores7 + 1;
        }
    }

    printf("Existem %d múltiplos de 7 e %d divisores de 7.", Multiplos7, Divisores7);
}
```


1.8 Preencher duas matrizes e somá-las (usando funções).

```
#include <stdio.h>
#define NL 10
#define NC 10
void PreencherMatriz( int M[NL][NC] , int Lin, int Col )
{
    int i,j;
    for( i = 0 ; i < Lin ; i++ )
    {
        for( j = 0 ; j < Col ; j++ )
        {
            scanf("%d",&M[i][j]);
        }
    }
}
void SomarMatrizes( int X[NL][NC] , int Y[NL][NC], int Z[NL][NC], int Lin, int Col )
{
    int i,j;
    for( i = 0 ; i < Lin ; i++ )
    {
        for( j = 0 ; j < Col ; j++ )
        {
            Z[i][j] = X[i][j] + Y[i][j];
        }
    }
}
main()
{
    int l,c,i,j;
    do{
        printf("Insira o número de linhas e colunas, com 0 < l < 10 e 0 < c < 10\n");
        scanf("%d %d",&l,&c);
    }while( l < 0 || l >= 10 || c < 0 || c >= 10 );
    int X[l][c], Y[l][c], Z[l][c];
    PreencherMatriz(X,l,c);
    PreencherMatriz(Y,l,c);
    SomarMatrizes(X,Y,Z,l,c);
}
```

- **Nota :**

- *#define NL 10* e *#define NC 10* são utilizados para evitar erros no processamento das matrizes.

1.9 Anular os elementos que não estão na diagonal de uma matriz.

```
#include <stdio.h>
#define NL 10
#define NC 10

void PreencherMatriz( int M[NL][NC] , int Lin, int Col )
{
    int i,j;
    for( i = 0 ; i < Lin ; i++ )
    {
        for( j = 0 ; j < Col ; j++ )
        {
            scanf("%d",&M[i][j]);
        }
    }
}

void TransformarMatriz( int X[NL][NC], int Lin, int Col )
{
    int i,j;
    for( i = 0 ; i < Lin ; i++ )
    {
        for( j = 0 ; j < Col ; j++ )
        {
            if ( i != j )
            {
                X[i][j] = 0;
            }
        }
    }
}

main()
{
    int lin,col,i,j;
    do{
        printf("Insira o número de linhas e colunas, com 0 < lin e col < 4 e lin = col.\n");
        scanf("%d %d",&lin,&col);
    }while( lin < 0 || lin >= 4 || col < 0 || col >= 4 || lin != col );
    int X[lin][col];
    PreencherMatriz(X,lin,col);
    TransformarMatriz(X,lin,col);
}
```

1.10 Determinar quantas vogais existem numa string.

1.10.1 Usando a estrutura condicional *if*.

```
#include <stdio.h>
main()
{
    char s[100];
    int x = 0;
    int a = 0, e = 0, i = 0, o = 0, u = 0;
    gets(s);

    while ( s[x] != '\0')
    {
        if ( s[x] == 'a' || s[x] == 'A' )
            a = a + 1;
        if ( s[x] == 'e' || s[x] == 'E' )
            e = e + 1;
        if ( s[x] == 'i' || s[x] == 'I' )
            i = i + 1;
        if ( s[x] == 'o' || s[x] == 'O' )
            o = o + 1;
        if ( s[x] == 'u' || s[x] == 'U' )
            u = u + 1;

        x++;
    }

    printf("A palavra tem %d a's, %d e's, %d i's, %d o's, %d u's.", a,e,i,o,u);
}
```

- **Nota :**

- Não é necessária a utilização de chavetas quando dentro dos *if*'s só existe uma linha de código.

1.10.2 Usando a estrutura *switch case*.

```
#include <stdio.h>

main()
{
    char s[100];
    int x = 0;
    int a = 0, e = 0, i = 0, o = 0, u = 0;
    gets(s);

    while ( s[x] != '\0')
    {
        switch ( s[x] )
        {
            case('a'): case('A'):
                a = a + 1;
                break;
            case('e'): case('E'):
                e = e + 1;
                break;
            case('i'): case('I'):
                i = i + 1;
                break;
            case('o'): case('O'):
                o = o + 1;
                break;
            case('u'): case('U'):
                u = u + 1;
                break;
        }

        x++;
    }
    printf("A palavra tem %d a's, %d e's, %d i's, %d o's, %d u's.", a,e,i,o,u);
}
```

- **Nota :**

- A estrutura utilizada designa-se por *switch case*. Esta funciona nos mesmos moldes que as estruturas condicionais.

1.11 Fazer uma cópia de uma ficheiro para outro.

```
#include <stdio.h>

main()
{
    FILE *file1;
    FILE *file2;
    char s[50];
    char u[50];
    int contador = 0;

    file1 = fopen("ficheiro.txt", "r");
    file2 = fopen("ficheirocópia.txt", "w");

    while ( !feof(file1) )
    {
        fscanf(file1, "%s", u);
        fprintf(file2, "%s\n", u);
    }

    fclose(file1);
    fclose(file2);
}
```

1.12 Calcula a média dos números de um ficheiro, e coloca os números pares desse mesmo ficheiro, no outro.

```
#include <stdio.h>

main()
{
    FILE *file1;
    FILE *file2;
    int x,y;
    int contador = 0;
    int soma = 0;

    file1 = fopen("ficheiro.txt", "r");
    file2 = fopen("ficheiro1.txt", "w");

    while ( !feof(file1) )
    {
        fscanf(file1, "%d", &x);

        if ( x % 2 == 0 )
        {
            fprintf(file2, "%d ", x);
        }

        soma = soma + x;
        contador = contador + 1;
    }

    printf("A média dos números do ficheiro é %0.3f", 1.0*soma/contador);

    fclose(file1);
    fclose(file2);
}
```

- **Nota :**

- O 1.0 no *printf* final é para garantir que será mostrado um valor real. Por vezes a utilização em simultâneo de variáveis de tipos diferentes pode originar problemas.