

## INTRODUÇÃO À PROGRAMAÇÃO

2ª Prova - 05/10/2010 - Prof. Marcus Ramos

1. Construir um programa que aceite como entrada, inicialmente, três números inteiros  $x_1$ ,  $x_2$  e  $x_3$ . Supor que  $x_1 < x_2 < x_3$ . Depois disso, o programa deve aceitar uma seqüência de quinze números inteiros. A função do programa é determinar a quantidade de valores que se encontram em cada uma das faixas (i) menor que  $x_1$ , (ii)  $x_1$  (inclusive) até  $x_2-1$ , (iii)  $x_2$  (inclusive) até  $x_3-1$  e (iv) maior ou igual a  $x_3$  (inclusive).

Exemplo: o usuário digita inicialmente 2, 25 e 48, e depois -3, 0, 1, -10, 8, 26, 32, 48, 54, 102, 11, 5, 29, 37, 44. Na saída o programa deve informar:

Faixa 1 (< 2): 4 ocorrências.

Faixa 2 (>=2 e <25): 3 ocorrências.

Faixa 3 (>=25 e <48): 5 ocorrências.

Faixa 4 (>=48): 3 ocorrências.

```
#include <stdio.h>
int main () {
    int i,n, f1=0, f2=0, f3=0, f4=0, x1, x2, x3;
    scanf ("%d",&x1);
    scanf ("%d",&x2);
    scanf ("%d",&x3);
    for (n=1;n<=15;n++) {
        scanf ("%d",&i);
        if (i<x1) f1++;
            else if (i<x2) f2++;
            else if (i<x3) f3++;
            else f4++;
    }
    printf ("Faixa 1 (<%d): %d ocorrências.\n",x1,f1);
    printf ("Faixa 2 (>=%d e <%d): %d ocorrências.\n",x1,x2,f2);
    printf ("Faixa 3 (>=%d e <%d): %d ocorrências.\n",x2,x3,f3);
    printf ("Faixa 4 (>=%d): %d ocorrências.\n",x3,f4);
}
```

2. Construir um programa que aceite como entrada um valor inteiro  $n$ , e gere na saída o resultado numérico da expressão:

$$(1^0/2)+(2^1/4)+(3^2/6)+\dots+(n^{n-1}/2*n)$$

```
#include <stdio.h>
int main () {
    float soma=0;
    int i,n;
    scanf ("%d",&n);
    for (i=1;i<=n;i++) {
        soma=soma+pow(i,i-1)/(2*i);
    }
    printf ("%f\n",soma);
}
```

3. Construir um programa que aceite como entrada duas seqüências de comprimento arbitrário de números binários (zeros e uns). Cada dígito de cada seqüência deverá ser

lido como um número inteiro. O término de cada seqüência deverá ser indicado por um valor negativo ou maior do que um. Na saída, o programa deverá gerar o número decimal que representa o número binário digitado, considerando que a primeira seqüência de dígitos representa a parte inteira do mesmo, e que a segunda seqüência representa a parte fracionária do mesmo.

Exemplo: considere o número binário  $x_1x_2\dots x_m.y_1y_2\dots y_n$ . As duas seqüências, nesse caso, são  $x_1x_2\dots x_m$  e  $y_1y_2\dots y_n$ . O valor decimal correspondente que deve ser informado na saída é aquele obtido pela fórmula:

$$x_1*2^{m-1} + x_2*2^{m-2} + \dots + x_m*2^0 + y_1*2^{-1} + y_2*2^{-2} + \dots + y_n*2^{-n}$$

Exemplo: a digitação das seqüências 1001 e 1101 (correspondente ao número binário 1001.1101) deve gerar na saída a impressão do número 9.8125 (resultado da avaliação da expressão  $1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 + 1*2^{-1} + 1*2^{-2} + 0*2^{-3} + 1*2^{-4}$ ).

```
#include <stdio.h>
int main () {
    float i=0,j=0;
    int k,pos;
    scanf ("%d",&k);
    while ((k==0) || (k==1)) {
        i=2*i+k;
        scanf ("%d",&k);
    }
    scanf ("%d",&k);
    pos=-1;
    while ((k==0) || (k==1)) {
        j=j+k*pow(2,pos);
        scanf ("%d",&k);
        pos=pos-1;
    }
    printf ("%f\n",i+j);
}
```

4. Construir um programa que, inicialmente, preenche um vetor de  $n$  posições (em ordem de índice crescente) com valores inteiros digitados pelo usuário no teclado. Considere que  $n$  é um número par definido através do comando #define. Em seguida, o programa deve imprimir os valores digitados, alternando entre o 1º, o último, o 2º, o penúltimo e assim por diante, até que todos os elementos do vetor sejam impressos na saída.

Exemplo: considere  $n=20$ . O usuário digita os valores 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 e 20. Na saída, a seguinte seqüência de valores deve ser impressa: 1, 20, 2, 19, 3, 18, 4, 17, 5, 16, 6, 15, 7, 14, 8, 13, 9, 12, 10, 11.

```
#include <stdio.h>
#define n 20
int main () {
    int v[n],i;
    for (i=0;i<n;i++) scanf ("%d",&v[i]);
    for (i=0;i<n/2;i++) printf ("%d\n%d\n",v[i],v[n-i-1]);
}
```