

INF01202

Algoritmos e Programação

Modalidade Ensino a Distância

Geração de Números Aleatórios

Tópico 11

Material desenvolvido pelas professoras:

- 2009/I

Maria Aparecida Livi, e Magda Bercht

Geração de números aleatórios

- A linguagem C dispõe de uma função para gerar números aleatórios, a função **rand**.
- A função **rand** gera números inteiros no intervalo de 0 até **RAND_MAX** (**RAND_MAX** é uma constante definida em **stdlib.h**).
- É uma função *sem parâmetros*, que existe em todos os compiladores e em todas as plataformas de desenvolvimento.
- Exemplo: **rand()**

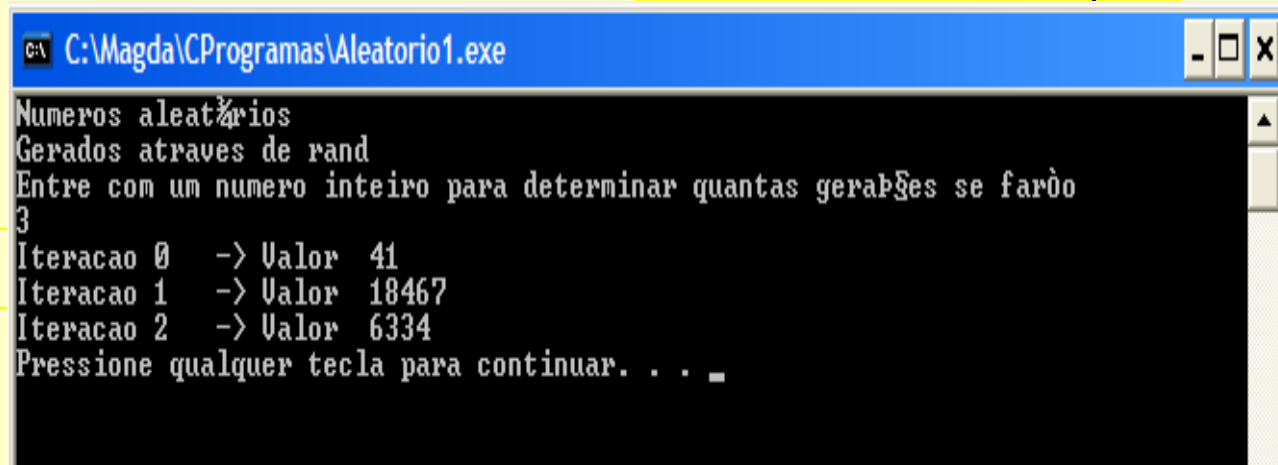
Função RAND: um exemplo de uso.
Programa que lê um inteiro n para
determinar quantos valores
aleatórios serão gerados.

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

int main ( ) // teste de numeros aleatórios
{
    int n,i,x;
    printf("Numeros aleatórios\nGerados atraves de rand\n");
    printf("Entre com um numero inteiro para determinar quantas gerações se farão\n");
    scanf("%d",&n);
    for (i = 0; i < n; i++)
    {
        x = rand(); // chamada de rand... x poderá ter um valor de 0 até RAND_MAX
        printf("Iteracao %d -> Valor %d \n",i,x);
    }
    system("pause");
    return 0;
}
```

Instância de execução

Para um n=3



```
C:\Magda\CProgramas\Aleatorio1.exe
Numeros aleatórios
Gerados atraves de rand
Entre com um numero inteiro para determinar quantas gerações se farão
3
Iteracao 0 -> Valor 41
Iteracao 1 -> Valor 18467
Iteracao 2 -> Valor 6334
Pressione qualquer tecla para continuar. . . _
```

Mas,
se quisermos uma geração de números em um intervalo
predeterminado??

Resposta:

Usa-se a idéia do resto da divisão inteira!!!

Ou seja,

seja um inteiro n , qualquer, $\neq 0$, então o resto da divisão do valor gerado pela `rand` e n , retornará sempre um inteiro pertencente ao intervalo $[0, n-1]$.

Exemplo:

.....

$n=5$;

`x= rand() % n;` // x poderá somente ter valores no intervalo $[0,4]$

.....

Exemplo de geração de valores em um intervalo $[0, n-1]$

```
#include <stdio.h>
int main ( )
{
    int n,l,x;
    printf("Entre com um numero inteiro limitador do intervalo de
    geração,\n maior que zero e menor que 200\n");
    scanf("%d",&n);
    printf("Geracao de 22 valores no intervalo de [0, %d ]\n",n-1);
    printf("\n" );
    for (l = 1; l < 22 ; l++)
    {
        x = rand()% n; // x terá um valor de 0 até n-1
        printf("Iteracao %d -> Valor %d \n",l,x);
    }
    system("pause");
    return 0;
}
```

Limite superior
do intervalo

C:\Magda\CProgramas\Aleatorio2.exe

```
100
Geracao de 22 valores no intervalo de [0, 99 ]
Iteracao 1 -> Valor 41
Iteracao 2 -> Valor 67
Iteracao 3 -> Valor 34
Iteracao 4 -> Valor 0
Iteracao 5 -> Valor 69
Iteracao 6 -> Valor 24
Iteracao 7 -> Valor 78
Iteracao 8 -> Valor 58
Iteracao 9 -> Valor 62
Iteracao 10 -> Valor 64
Iteracao 11 -> Valor 5
Iteracao 12 -> Valor 45
Iteracao 13 -> Valor 81
Iteracao 14 -> Valor 27
Iteracao 15 -> Valor 61
Iteracao 16 -> Valor 91
Iteracao 17 -> Valor 95
Iteracao 18 -> Valor 42
Iteracao 19 -> Valor 27
Iteracao 20 -> Valor 36
Iteracao 21 -> Valor 91
Pressione qualquer tecla para continuar. . .
```

A seguir um programa que preenche uma matriz 8X8 com valores aleatórios no intervalo [1,200] e mostrar a soma dos elementos da diagonal principal

- Exemplo Comentado -

Gerar uma matriz 8X8 de valores aleatórios no intervalo [1,200] e mostrar a soma dos elementos da diagonal principal

```
// parte do programa para a geração dos valores da matriz M
#define LIMINTERVALO 200

.....

int lin, col;

int M[8][8];

for (lin = 0; lin < 8; lin++) //gerando os valores aleatórios no [1,200]
    for (col = 0; col < 8; col++)
        m[lin][col] = 1 + rand( ) % LIMINTERVALO; // atenção: 1 + rand()%LIm...

// mostrar a matriz gerada
for (i = 0; i < MAX; i++)
{
    printf("\n");
    for (j = 0; j < MAX; j++)
        printf("%5d", tabela[i][j]);
}
printf("\n");
```

//continua ...

- Exemplo Comentado -

// parte do programa para a soma dos elementos da diagonal principal

soma = 0; //inicializa em 0, pois eh soma!

for (lin = 0; lin < 8; lin++) //calculando a soma dos valores da diagonal

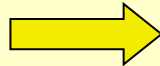
for (col = 0; col < 8; col++)

if (lin == col)

soma= soma + m[lin] [col];

printf ("\n\n A soma dos elementos da diagonal principal vale: %d",
soma);

Saída resultante:



```
C:\Magda\CProgramas\ALeatorio3.exe
Matriz gerada com valores entre 1 e 200
 42   68  135  101  170  125   79  159
163   65  106  146   82   28  162   92
196  143   28   37  192    5  103  154
 93  183   22  117  119   96   48  127
172  139   70  113   68  100   36   95
104   12  123  134   74   65  142  112
 54   69  148   45   63  158   38   60
124  142  130  179  117   36  191   43

A soma dos elementos da diagonal principal vale: 466
Pressione qualquer tecla para continuar. . .
```


Então,

Como gerar um número aleatório entre 1 e n?

Como a função **rand()** devolve sempre um número entre os limites 0 e **RAND_MAX**,
é necessário obter um número entre 0 e (n-1), faz-se :

rand() % n

e

para se obter valores no intervalo [1 , 10] ,
será necessário somar 1 ao valor obtido, como:

sorteado = 1 + rand() % n.

O conjunto de números aleatórios gerados é sempre igual.

Como solucionar esse problema?

A função `rand()` parte sempre da mesma semente para a geração dos aleatórios. Assim, a semente deve ser alterada para o sorteio ficar diferente, ou seja, gerar uma sequência de valores diferentes.

Uma maneira é variar a semente mudar em cada execução do programa. E, pode-se usar um valor que é gerado através do relógio do computador.

Primeiro, declara-se uma variável, seja, `ultime` do tipo **`long int`**, seja:

```
long ultime;
```

Segundo, essa variável é inicializada com o número de segundos desde 01/01/1970, como,

```
time(&ultime);
```

Terceiro, efetua-se a carga inicial da semente através da função **`srand`**, da seguinte forma:

```
srand((unsigned) ultime);
```

Exemplo: o exemplo anterior com gerações diferentes

Gerar uma matriz 8 X8 de valores aleatórios no intervalo [1,200] e mostrar a soma dos elementos da diagonal principal

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>      // biblioteca para funções de tempo, necessária
                        para usar time
#define LIMINTERVALO 200

int main ( )
{
    int lin, col,i,j,soma;
    int m[8][8];

    long int ultime;
    time(&ultime);
    srand( (unsigned) ultime);
```

ultime, variável iniciada com o número de segundos, pela função time

Função srand que inicializa a semente de sorteio

.....
continua....

Exemplo: gera sequencias aleatórias diferentes

```
for (lin = 0; lin < 8; lin++) // gerando os valores aleatórios no [1,200]
    for (col = 0; col < 8; col++)
        m[lin] [col] = 1 + rand( ) % LIMINTERVALO;// atenção: 1 + rand()%Lim...
// mostrar a matriz gerada
printf("Matriz gerada com valores entre 1 e 200\n");
for (i = 0; i < 8; i++)
{
    printf("\n");
    for (j = 0; j < 8; j++)
        printf("%5d", m[i][j]);
}
printf("\n");
soma = 0; //inicializa em 0, pois eh soma!
for (lin = 0; lin < 8; lin++) //calculando a soma dos valores da diagonal
    for (col = 0; col < 8; col++)
        if ( lin == col )
            soma= soma + m[lin] [col];
printf ("\n\n A soma dos elementos da diagonal principal vale: %d \n", soma);
system("pause");
return 0;
}
```

Exemplo de execuções do mesmo programa

```
C:\Magda\CProgramas\ALeatorio3.exe
Matriz gerada com valores entre 1 e 200

 75  104  170  160  195  161   5  15
 67  133   10  142  174  104  34  102
150  157  135  192  193   25 168  162
 53  199  147  122  128   84 112  107
187   86   2   62  111   81 185   98
 81   26  199  151   26  199   16  159
138  126  117  174   54  118  106  187
 31   60  159   24  126   14  124  199

A soma dos elementos da diagonal principal vale: 1080
Pressione qualquer tecla para continuar. . . _
```

↑
**Atenção: as 2
execuções
produzem valores
diferentes, pois a
semente de
geração é alterada
a cada execução!!**



```
C:\Magda\CProgramas\ALeatorio3.exe
Matriz gerada com valores entre 1 e 200

 54  135   30   24   70  110   1  131
100   53   20   51   11  166  21   72
 78  163   90  110  104  142  184   55
 30   65  138  158   31   56  173  169
155   46  108   70   34   37   63   58
182  138  129  197   84   72  162   98
103   12   82   30   22  105  112   79
198  197  162  127  162   19  166   72

A soma dos elementos da diagonal principal vale: 645
Pressione qualquer tecla para continuar. . .
```