



Algoritmos e Programação II

Prof. Joilson dos Reis Brito

Agenda da Aula

- Como gerar números aleatórios
- Preencher Matrizes com números aleatórios

Números Aleatórios

Para resolver alguns problemas computacionais é necessário manipular uma quantidade maior de valores para teste. Como inserir manualmente muitos números se torna uma tarefa cansativa, é possível usar uma função que gera automaticamente números para preencher estruturas compostas.

Gerar sequências de números aleatórios é um problema bastante comum na programação.

Para gerar um número aleatório (randômico) em linguagem C podemos usar a função **rand** pertencente à biblioteca `stdlib.h`

A função rand

A função rand produz um valor aleatório na faixa entre 0 e a constante RAND_MAX. O valor dessa constante é definido na biblioteca stdlib.h

Muitas vezes necessitamos gerar valores dentro de determinada faixa.

Vamos supor que a faixa de valores desejada esteja entre o valor mínimo zero e o valor máximo 100, precisamos pegar o resto da divisão de rand() por 100

```
int X;
```

```
X = rand() % 100;
```

```
printf("Numero aleatorio: %d", X);
```

Exemplo 1 - rand()

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int I;
6
7      printf("Gerando 5 valores aleatorios:\n\n");
8
9      for (I = 0; I < 5; I++)
10     {
11         /* gerando valores aleatórios entre zero e 100 */
12         printf("%d ", rand()%100);
13     }
14     return 0;
15 }
```

Exemplo 1 - rand()

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int I;
6
7      printf("Gerando 5 valores aleatorios:\n\n");
8
9      for (I = 0; I < 5; I++)
10     {
11         /* gerando valores aleatórios entre zero e 100 */
12         printf("%d ", rand()%100);
13     }
14     return 0;
15 }
```

Gerando 5 valores aleatorios:

41 67 34 0 69

Process exited with return value 0

Press any key to continue . . . ■

Exemplo 1 - rand()

Agora execute o código novamente e observe o que acontece!

Porque os números se repetem em todas as execuções?

Para gerar uma sequência de valores diferentes a cada execução é necessário utilizar a função **srand** que inicializa a função rand com um valor “semente” de tal forma que esta semente seja um valor diferente a cada execução do programa

Função srand()

A função srand a fim de inicializar a função rand com um valor “semente” para que se produza um valor aleatório na faixa determinada.

A função srand recebe um argumento do tipo inteiro sem sinal.

```
srand (time (NULL) ) ;
```

srand(time(NULL)) objetiva inicializar o gerador de números aleatórios com o valor da função time(NULL). Este por sua vez, é calculado como sendo o total de segundos passados desde 1 de janeiro de 1970 até a data atual.

Desta forma, a cada execução o valor da "semente" será diferente

Exemplo 2 - srand() e rand()

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  int main()
5  {
6      int I;
7
8      printf("Gerando 5 valores aleatorios:\n\n");
9
10     srand(time(NULL));
11
12     for (I = 0; I < 5; I++)
13     {
14         /* gerando valores aleatórios entre zero e 100 */
15         printf("%d ", rand()% 100);
16     }
17     return 0;
18 }
```

Biblioteca da função time, usada pela função srand na linha 10.

Exemplo 2 - srand() e rand()

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  int main()
5  {
6      int I;
7
8      printf("Gerando 5 valores aleatorios:\n\n");
9
10     srand(time(NULL));
11
12     for (I = 0; I < 5; I++)
13     {
14         /* gerando valores aleatórios entre zero e 100 */
15         printf("%d ", rand()% 100);
16     }
17     return 0;
18 }
```

Gerando 5 valores aleatorios:

84 39 38 86 61

Process exited with return value 0
Press any key to continue . . . ■

Agora para preencher um vetor com números aleatórios, basta fazer:

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define TAMANHO 100
```

```
int main()
{
    int Numeros[TAMANHO],I;
    srand(time(NULL));
    for(I=0;I<TAMANHO;I++)
        Numeros[I]=rand()%100;

    for(I=0;I<TAMANHO;I++)
    {
        printf(" %2d ",Numeros[I]);
    }
    return 0;
}
```

No exercício 2 lista 2 utilize a geração aleatória de números para preencher o vetor e altere o tamanho do vetor para ver como o programa vai classificar o vetor rapidamente, mesmo para vetores com muitos elementos