A função rand()

Disciplina: Algoritmos e Programação
Curso: Engenharia de Computação

Professora: Mariza Miola Dosciatti
mariza@utfpr.edu.br

Função rand()

- Gerar sequências de números aleatórios é um problema bastante comum em programação.
- Para gerar um número aleatório (randômico) em linguagem **C** podemos usar a **função rand** pertencente à biblioteca **stdlib.h**.
- Quando esta **função** é chamada, ela produz um valor aleatório na faixa entre **0** e a constante **RAND_MAX** que tem valor **32767**. O valor desta constante encontra-se definida no arquivo stdlib.h.

Função rand() (cont.)

- Agora observe o seguinte: se o programa a seguir for executado várias vezes, o valor gerado a cada uma das vezes será exatamente a mesma. Por que isto ocorre?

```
#include <stdio.h>
#include <stdlib.h>//biblioteca necessária para rand()e
                   //RAND_MAX

int main(void)
{
        int x;

        x = rand();

        printf("%d\n", x);

        return 0;
}
```

Função rand() (cont.)

- Para a função **rand()** funcionar adequadamente ela precisa de um valor inicial chamado de “semente”. Se nenhum valor é passado como semente, **rand()** assume o valor 1 e assim gera sempre a mesma sequência de números a cada execução.

Função rand() (cont.)

- Como gerar valores diferentes?
  - Para gerar valores diferentes a cada execução é necessário utilizar a função **srand()** que inicializa a função **rand()** com um valor "semente", de tal forma que este valor seja diferente a cada execução do programa.

Função rand() (cont.)

- Como gerar valores diferentes?
  - A biblioteca time.h possui a função **time()** cujo resultado é o número de segundos transcorridos desde 1 de janeiro de 1970.
  - A função **time()** é usada para inicializar o gerador de números aleatórios.
    - O parâmetro da função time() pode ser um *ponteiro nulo (NULL)*, caso em que o parâmetro não é usado.
  - A cada execução o valor "semente" de **rand()** será diferente.

Para gerar valores entre 0 e RAND_MAX

```c
#include <stdio.h>
#include <stdlib.h>//biblioteca necessária para rand()e RAND_MAX
#include <time.h>

int main(void)
{
	int x;

	srand(time(NULL));

	//Gerando valores aleatórios entre 0 e RAND_MAX
	x = rand();
	printf("%d\n", x);

	return 0;
}
```

Para gerar valores entre 0 e limite

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

int main(void)
{
	int x, limite=10;

	srand(time(NULL));

	x = rand() % (limite + 1);
	printf("%d\n", x);

	return 0;
}
```

Para gerar valores entre 1 e limite

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

int main(void)
{
	int x, limite=10;

	srand(time(NULL));

	x = rand() % limite + 1;
	printf("%d\n", x);

	return 0;
}
```

- **rand() % limite** gera valores aleatórios de 0 a 9.
- Acrescentar 1 vai gerar valores de 1 a limite.

Para gerar valores entre em uma faixa de valores

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

int main(void)
{
	int x, limInf=5, limSup=15;

	srand(time(NULL));

	x = rand() % ((limSup - limInf) + 1) + limInf;
	printf("%d\t", x);

	return 0;
}
```

- **(limSup – limInf)** vai resultar em 10
- **rand() % (10 + 1)** gera valores de 0 a 10.
- Somar **limInf** permite alcançar a faixa desejada (5 a 15).

Para gerar valores positivos e negativos entre 1 e -1

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

int main(void)
{
      int x;

      srand(time(NULL));

      x = rand() % 3 - 1;
      printf("%d\n", x);

      return 0;
}
```

- **rand() % 3** gera valores de 0 a 2.
- Subtraindo 1, gera valores na faixa entre -1 e 1.

Para gerar valores float entre 0 e 1

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

int main(void)
{
    float x;

    srand(time(NULL));

    x = (float)(rand()) / RAND_MAX;
    printf("%.1f\n", x);

    return 0;
}
```

Para gerar valores positivos e negativos

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

int main(void)
{
    int x, limPositivo=5, limNegativo=5;

    srand(time(NULL));

    x = rand() % (limPositivo + limNegativo + 1) - limNegativo;
    printf("%d\n", x);

    return 0;
}
```