
Gerando valores aleatórios em C



Valores aleatórios em C com a 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 **funções** pertencente à biblioteca **stdlib.h**.
- Quando esta **função** é chamada ela produz um valor aleatório na faixa entre **0** e a constante **RAND_MAX**.

A função **rand()** é usada para gerar números pseudoaleatórios.



Retorna um número inteiro entre 0 e **RAND_MAX** (constante definida na biblioteca `<stdlib.h>`).

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

```
main(){
```

```
    int i;
```

```
    printf("Gerando 10 valores aleatorios:\n\n");
```

```
    for (i = 0; i < 10; i++){
```

```
        /* gerando valores aleatórios entre zero e 100 */
        printf("%d ", rand() % 100);
    }
```

```
    getch();
}
```

rand() % 100 gera um número aleatório entre 0 e 99.

rand() gera um número inteiro não negativo;

Operador % faz a operação módulo (resto da divisão), limitando os valores entre 0 e 99.

Exemplos:

- Se **rand()** gerar 12345, então $12345 \% 100$ é **45**.
- Se **rand()** gera 56789, então $56789 \% 100$ é **89**.
- Se **rand()** gerar 2147483647 (o maior valor possível para **RAND_MAX**), então $2147483647 \% 100$ é **47**.

```
C:\Users\joomla\Desktop\sequencia com dez numeros aleatorios.exe
Gerando 10 valores aleatorios:
41 67 34 0 69 24 78 58 62 64
```



rand()

Usada para gerar números **pseudoaleatórios**. Cada vez que você chama rand(), ela retorna um número inteiro entre **0** e **RAND_MAX**.

Os números gerados por essa função são chamados de pseudoaleatórios porque são gerados por um **algoritmo determinístico**.

*Algoritmo determinístico é aquele que, **dado um determinado estado inicial e uma entrada específica**, produz sempre a mesma saída e segue o mesmo caminho durante sua execução.*

A sequência de números produzida por **rand()** depende de uma semente (**seed**) inicial.

Assim, utilizando-se sempre a mesma semente, a sequência será sempre a mesma.

Este comportamento é útil para testes e depuração, mas não é adequado para aplicações que requerem números verdadeiramente aleatórios, como criptografia.



Para gerar uma sequência de **valores diferentes a cada execução** é necessário utilizar a função **srand**.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h> // necessário p/ as funções rand() e srand()
#include <time.h>   //necessário p/ função time()
```

```
main(){
    int i;
    srand(time(NULL));

    for (i = 0; i < 10; i++) {
        printf("%d ", rand() % 100);
    }
}
```

srand - inicializa a função **rand** com um valor “**semente**” de forma que esta semente seja um valor diferente a cada execução do programa. Isso acarreta em uma **sequência de valores diferente a cada execução**.

Sem inicializar o gerador com **srand()**, o **rand()** usará uma semente (*seed*) padrão, resultando na mesma sequência de números em cada execução do programa.

A função **time()** retorna a hora atual em segundos desde o **Epoch** (00:00:00 UTC, 1 de janeiro de 1970).

Quando chamada com o argumento NULL, **time(NULL)** retorna esse valor como um **time_t**, que é essencialmente um inteiro.

Exercícios



1) Analise os seguintes algoritmos e diga o que será impresso ao serem executados

a)

$B \leftarrow 5$
$A \leftarrow 0$
$A < B$
$A \leftarrow A + 1$
$B \leftarrow B - 1$
Escreva A, B

1b)

$B \leftarrow 5$
$A \leftarrow 0$
Escreva A, B
$A \leftarrow A + 1$
$B \leftarrow B - 1$
$B < A$



Exercício 2 – Chute certo



Escreva um programa na Linguagem C na qual o computador escolhe um número de 0 a 100 e o jogador tenta acertar o número escolhido.

O computador irá dando dicas se o número escolhido é maior ou menor que o chutado pelo jogador. O jogador terá 5 chances.

Caso o jogador não acerte deverá ser informado o número escolhido. O jogo acaba caso acabe as chances ou jogador acertar o numero escolhido.





Exercício 3 - Palpiteiro

Escreva um Programa em C na qual o computador irá dar palpites de dezenas para a Mega Sena e Loto. Deverá ter a seguinte tela:

PROGRAMA PARA PALPITES

Digite: 1-LOTO 2-Mega-Sena 3-Dupla-Sena

[Deseja sair do Programa ? (S/N)]

Caso o usuário digite 1 o computador irá dar 5 palpites para a Loto, se o usuário digitar 2 deverá dar 6 palpites para a Mega Sena. Caso digite 3, deverão ser lançados 12 palpites para a Dupla Sena. *(Deverá ser feito um teste para aceitar somente os valores 1, 2 e 3.)*

Após dado os palpites o programa deverá perguntar se o usuário deseja sair do programa ou receber novos palpites. Caso o usuário deseje sair do programa deverá ser informado quantas vezes o programa foi utilizado.



Gerando valores aleatórios em C