

Introdução à programação C/C++

aula prática 2¹

1 Exercícios

1.1 Exercício 1

Escreva um programa que imprima a figura abaixo. O número de linhas e colunas deve ser pedido ao usuário.

```
*****
*****
*****
*****
```

1.2 Exercício 2

Escreva um programa que imprima uma figura como a mostrada abaixo. O número de linhas da figura deve ser pedido ao usuário.

```
*
**
***
****
*****
*****
```

1.3 Exercício 3

Escreva um programa que imprima uma figura como a mostrada abaixo. O número de linhas da figura deve ser pedido ao usuário.

```
*****
*****
****
***
**
*
```

1.4 Exercício 4

Escreva o programa 1 e verifique os resultados. Este programa mostra como podemos gerar números inteiros randômicos em um intervalo. A função `srand` serve para inicializar o gerador de

1 Este material é praticamente uma reprodução do produzido pelo Prof. Adriano Joaquim de Oliveira Cruz, disponível em <http://equipe.nce.ufrj.br/adriano/c/praticas.htm>

números aleatórios. A função `rand` gera um número inteiro aleatório entre 0 e `RAND_MAX` (uma constante definida em `stdlib.h`).

```
#include<stdio.h>
#include<time.h>
#include<stdlib.h>
int main( void ) {
    int i, n;
    int max = 100;

    /* - srand inicia o gerador de números aleatórios
       só precisa aparecer uma vez no início do programa
       - time(NULL) retorna o número de segundos desde 01/01/1970
       - srand deve receber um inteiro para iniciar o gerador de números
         randômicos
       - costuma-se usar o número de segundos fornecido por time(NULL)
    */

    srand(time(NULL));

    printf("RAND_MAX: %d\n", RAND_MAX);
    for (i = 0; i < 10; i++){
        n = rand();          //gera um número inteiro entre 0 e RAND_MAX
        n = n % max;         //passa este número para o intervalo [0,max)
        printf("%d\n",n);
    }
    return 0;
}
```

1.5 Exercício 5

Vamos fazer um teste inicial para verificar se o gerador de números aleatórios tem vícios. Vamos testar se ele gosta mais dos números pares ou dos ímpares. Escreva um programa que gere 1.000.000 números aleatórios entre 0 e `RAND_MAX` e calcule quantos destes são pares e quantos são ímpares. Calcule a frequência (em percentagem) com que cada um destes tipos foi gerado. O que você pode dizer após este pequeno experimento. Rode o programa diversas vezes e verifique se os resultados levam a mesma conclusão.

1.6 Exercício 6

Neste exercício é obrigatório o uso do comando `switch`

Vamos avançar um pouco mais, para isto vamos criar um dado eletrônico. Será que o dado é honesto ou tem alguma tendência? Escreva um programa que gere 6.000.000 números aleatórios entre 1 e 6 e calcule a frequência (em percentagem) com que cada um destes números foi gerado. Observe que o gerador de números randômicos gera números entre 0 e um número escolhido e isto pode ser um problema. Como fazer que ele gere números entre 1 e 6? O que você pode dizer sobre o dado?

1.7 Exercício 7

Vamos usar o gerador de números aleatórios para calcular π . Para isto comece considerando a Figura 1. Escreva um programa que gere 1.000.000 de pontos cujas coordenadas devem estar entre $(0 \leq x, y \leq 1.0)$. Conte o número de pontos (*dentro*) que estão dentro do quarto de círculo (Figura 1) de raio 1.0 e centro na origem.

(a) Imprima o resultado da seguinte conta:

$$X = \frac{4 \times \text{dentro}}{1000000}$$

(b) Faça o experimento outra vez, agora com 9.999.999 pontos.

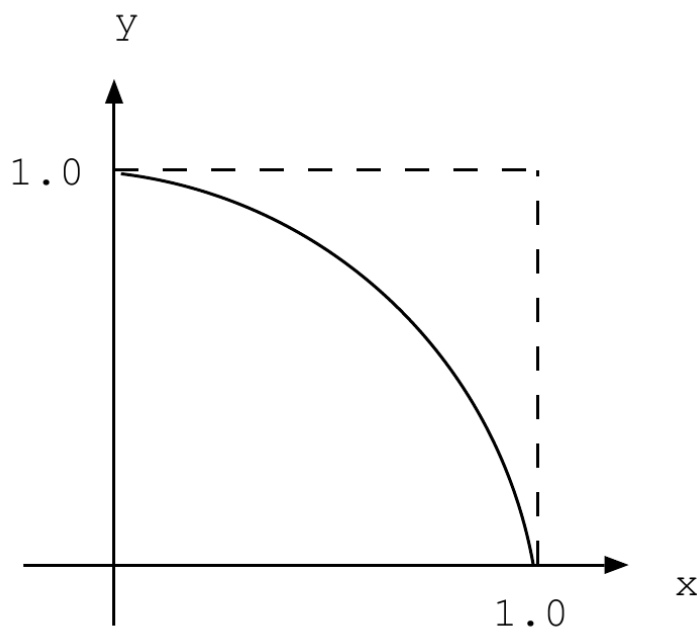


Figura 1