



Aula Prática P-04

- * Todos os exercícios que envolvem programas devem ser resolvidos através de programas em C/C++.
 - * A entrega será feita até às 23h55 do dia da aula prática no Moodle, sem zipar (entregue apenas o código fonte)
 - * Inclua seu número de matrícula, nome e turma em um comentário no início de cada arquivo com código fonte.
 - * Você só pode utilizar conhecimento prévios à aula para resolver o exercício. Caso use uma matéria que ainda não foi dada sua nota será penalizada.
 - * Códigos que não compilam serão zerados.
-

Questão 01

Crie uma função que retorna o fatorial $n!$ de um número n passado por parâmetro.

Use o seguinte protótipo:

```
long long fat(int n);
```

Lembre-se que $n! = 1 \times 2 \times 3 \times \dots \times n$. Por exemplo, $6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6$.

Lembre-se também que devemos utilizar o formato "%lld" para imprimir um `long long`.

Em seguida, implemente a função `main()` para ler o valor de `n` e imprimir o resultado de `fat(n)`.

Exemplo de execução (valores digitados pelo usuário destacados em azul):

```
1 Digite o valor de n: 6
2
3 6! = 720
```

Questão 02

Um número natural é primo se ele possui apenas dois divisores distintos. Assim, um número n maior que 1 ($n > 1$) é primo se n for divisível apenas por si próprio n e por 1.

Crie uma função que retorna 1 se o número passado como argumento for primo e 0 caso contrário.

Sua função deve seguir o protótipo:

```
int is_prime(int n);
```

Implemente também a função `main()` para ler o valor de `n` e imprimir se `n` é primo ou não.

Exemplos de execução (valores digitados pelo usuário destacados em azul):

```

1 Digite um número: 29881
2
3 29881 é um número primo!

```

```

1 Digite um número: 16
2
3 16 NÃO é um número primo!

```

Quando o programa estiver funcionando, note que vários números entre 1 e n não precisam, por um motivo ou outro, serem testados. Tente perceber se é possível tornar sua função mais eficiente evitando o teste destes números.

Questão 03

Faça um programa que desenha um trapézio isósceles de bases x e y usando asteriscos. O usuário deve informar os valores de x e y , que devem ser **pares** e tais que $x < y$.

Certifique-se de que o usuário digitou valores válidos.

Exemplo de execução (valores digitados pelo usuário destacados em azul):

```

1 Digite os valores de x e y: 3 6
2 Apenas números pares são aceitos.
3
4 Digite os valores de x e y: 10 8
5 Erro: x deve ser menor que y.
6
7 Digite os valores de x e y: 4 18
8
9      ****
10     *
11    *
12   *
13  *
14 *
15 *
16 *

```

Questão 04 (Opcional)

Desde a aula de ontem, *Bart Simpson* continua tentando aprender a jogar xadrez. Ele aprendeu como uma Torre se move, mas tem dificuldade em saber para qual direção ele pode mover um **Bispo**. Sabendo que um tabuleiro de xadrez é composto por 8 linhas e 8 colunas, e que o **Bispo** se move nas diagonais:

- Escreva um programa que solicite ao *Bart* o número da linha e da coluna que indicam a posição do **Bispo**. O programa deve imprimir quais são os possíveis movimentos.
- Utilize "-" para indicar uma casa para a qual o **Bispo** não pode ser movido e "x" para indicar uma casa para a qual ele pode ser movido.

```
1 Movimentos de um Bispo no xadrez!
2 Digite a linha em que o Bispo se encontra: 6
3 Digite a coluna em que o Bispo se encontra: 3
4
5 Movimentos possíveis:
6
7      1  2  3  4  5  6  7  8
8      -----
9  1 | -  -  -  -  -  -  -  x
10 2 | -  -  -  -  -  -  x  -
11 3 | -  -  -  -  -  x  -  -
12 4 | x  -  -  -  x  -  -  -
13 5 | -  x  -  x  -  -  -  -
14 6 | -  -  x  -  -  -  -  -
15 7 | -  x  -  x  -  -  -  -
16 8 | x  -  -  -  x  -  -  -
```