



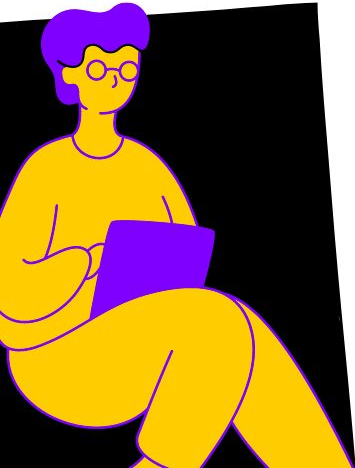
REVISÃO E EXERCÍCIOS FUNÇÕES FOR E RAND

PROJETO SIM, ELAS
PODEM!

ESTRUTURA DE REPETIÇÃO - FOR

EXECUTA A REPETIÇÃO DE UM BLOCO DE INSTRUÇÕES **PARA** UM NÚMERO INICIAL, ATÉ UM NÚMERO FINAL. SENDO ESSE VALOR INCREMENTADO A CADA LOOP

```
for(valor_inicial;condição_final;valor_increento)
{
    instruções;
}
```



FUNÇÃO RAND E SRAND

- Serve para gerar/sortear valores randômicos (aleatórios) dentro de um programa.

`a = rand()% 100;` a está no intervalo de 0 a 99

`b = rand()% 100 + 1;` b está no intervalo de 1 a 100

`c= rand()% 30 + 1985;` c está no intervalo de 1985 a 2014

- Para gerar sempre valores aleatórios, torna-se necessário mudar a “semente”

- Para mudar a “semente” sempre que rodarmos o código, usamos a função:

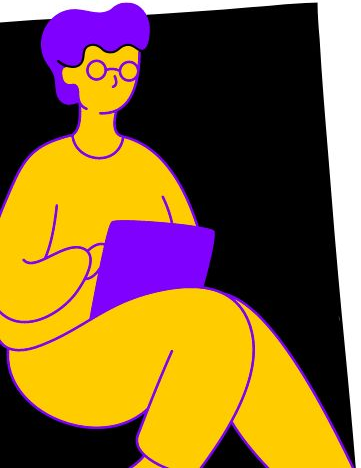
```
srand(time(NULL))
```



EXEMPLO

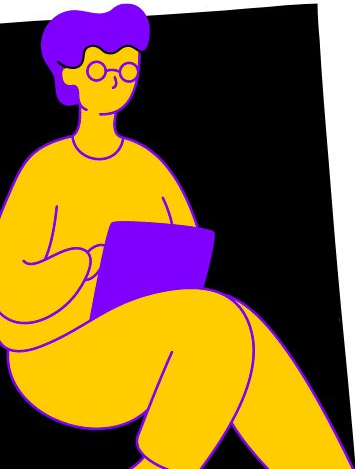
```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int a, b;
    srand(time(NULL));
    a = rand()% 100; // 0 a 99
    b = rand()% 10 + 1; // 1 a 10
    cout << a << endl;
    cout << b << endl;
}
```



EXERCÍCIO 1

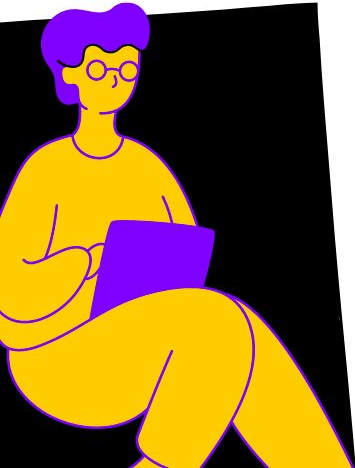
ESCREVA UM PROGRAMA QUE LEIA
DOIS NÚMEROS, A & B, E INFORME OS
NÚMEROS ENTRE ELES



EXEMPLO DE ENTRADA	EXEMPLO DE SAÍDA
1 10	2 3 4 5 6 7 8 9
13 17	14 15 16

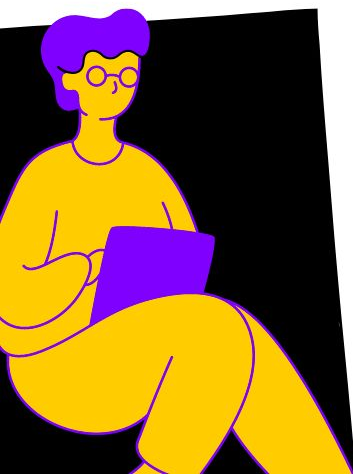
EXERCÍCIO 2

FAÇA UM PROGRAMA QUE
IMPRIMA O SEGUINTE PADRÃO



```
*  
**  
***  
****  
*****  
******  
*******  
********  
 *********  
          **  
         ***  
        ****  
       *****  
      ******  
     *******  
    *******  
   *******  
  *******  
 *~~~~~*
```

EXERCÍCIO 3



Sequências Crescentes

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 2

Este programa deve ler uma variável inteira **X** inúmeras vezes (deve parar quando o valor no arquivo de entrada for igual a zero). Para cada valor lido imprima a sequência de 1 até **X**, com um espaço entre cada número e seu sucessor.

Obs: cuide para não deixar espaço em branco após o último valor apresentado na linha ou você receberá **Presentation Error**.

Entrada

O arquivo de entrada contém vários números inteiros. O último número no arquivo de entrada é 0.

Saída

Para cada número **N** do arquivo de entrada deve ser impressa uma linha de 1 até **N**, conforme o exemplo abaixo. Não deve haver espaço em branco após o último valor da linha.

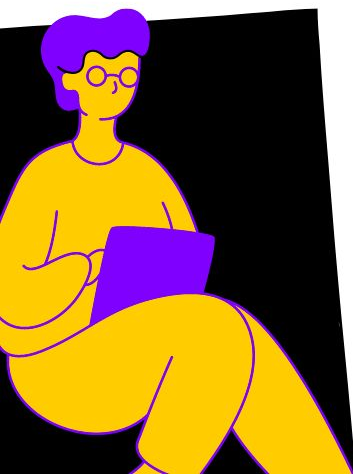
Exemplo de Entrada

5
10
3
0

Exemplo de Saída

1 2 3 4 5
1 2 3 4 5 6 7 8 9 10
1 2 3

EXERCÍCIO 4



Tipo de Combustível

Adaptado por Neillor Tonin, URI  Brasil

Timelimit: 1

Um Posto de combustíveis deseja determinar qual de seus produtos tem a preferência de seus clientes. Escreva um algoritmo para ler o tipo de combustível abastecido (codificado da seguinte forma: 1.Álcool 2.Gasolina 3.Diesel 4.Fim). Caso o usuário informe um código inválido (fora da faixa de 1 a 4) deve ser solicitado um novo código (até que seja válido). O programa será encerrado quando o código informado for o número 4.

Entrada

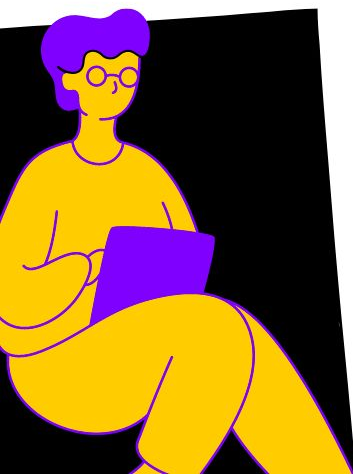
A entrada contém apenas valores inteiros e positivos.

Saída

Deve ser escrito a mensagem: "MUITO OBRIGADO" e a quantidade de clientes que abasteceram cada tipo de combustível, conforme exemplo.

Exemplo de Entrada	Exemplo de Saída
8	MUITO OBRIGADO
1	Alcool: 1
7	Gasolina: 2
2	Diesel: 0
2	
4	

EXERCÍCIO 5



URI Online Judge | 1143



Quadrado e ao Cubo

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Escreva um programa que leia um valor inteiro N ($1 < N < 1000$). Este N é a quantidade de linhas de saída que serão apresentadas na execução do programa.

Entrada

O arquivo de entrada contém um número inteiro positivo N .

Saída

Imprima a saída conforme o exemplo fornecido.

Exemplo de Entrada	Exemplo de Saída
5	1 1 1 2 4 8 3 9 27 4 16 64 5 25 125

EXERCÍCIO 6



URI Online Judge | 1095

Sequencia IJ 1

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Você deve fazer um programa que apresente a sequencia conforme o exemplo abaixo.

Entrada

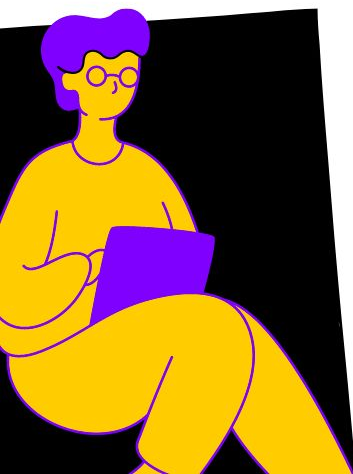
Não há nenhuma entrada neste problema.

Saída

Imprima a sequencia conforme exemplo abaixo

Exemplo de Entrada	Exemplo de Saída
	I=1 J=60
	I=4 J=55
	I=7 J=50
	...
	I=? J=0

EXERCÍCIO 7



URI Online Judge | 1133

Resto da Divisão

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Escreva um programa que leia 2 valores X e Y e que imprima todos os valores entre eles cujo resto da divisão dele por 5 for igual a 2 ou igual a 3.

Entrada

O arquivo de entrada contém 2 valores positivos inteiros quaisquer, não necessariamente em ordem crescente.

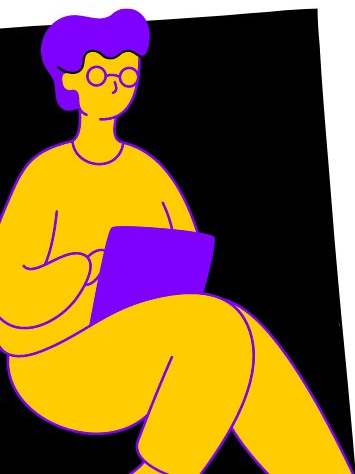
Saída

Imprima todos os valores conforme exemplo abaixo, sempre em ordem crescente.

Sample Input	Sample Output
10	12
18	13
	17

EXERCÍCIO 8

Faça um programa em C++ que represente um jogo onde o jogador deve tentar adivinhar 3 números, sendo que o primeiro número está no intervalo de 1 a 5, o segundo número está no intervalo de 6 a 10 e o terceiro número está no intervalo de 11 a 15. A entrada dos números deve ser feita de acordo com a sequência dos intervalos, onde o primeiro palpite deve ser de um número entre 1 e 5, e assim por diante. O programa deve calcular o número de acertos do jogador e imprimir na tela. Cada vez que o jogador acerta o número sorteado, é somado 1 ao número de acertos.

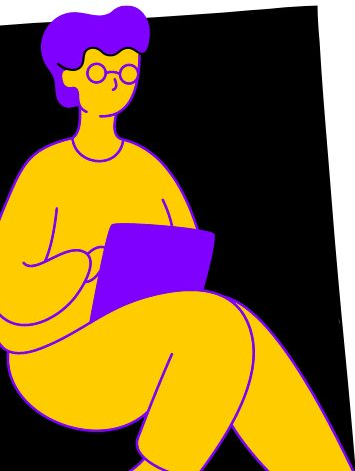


EXEMPLOS DE ENTRADA	EXEMPLOS DE SAÍDA
Entre com um número que esteja no intervalo de 1 a 5: 2 Número sorteado: 1	Seu número de acertos foi 1
Entre com um número que esteja no intervalo 6 a 10: 7 Número sorteado: 7	
Entre com um número que esteja no intervalo de 11 a 15: 13 Número sorteado: 14	

EXERCÍCIO 9

Faça um programa em C++ que simule um jogo de dados com as seguintes características:

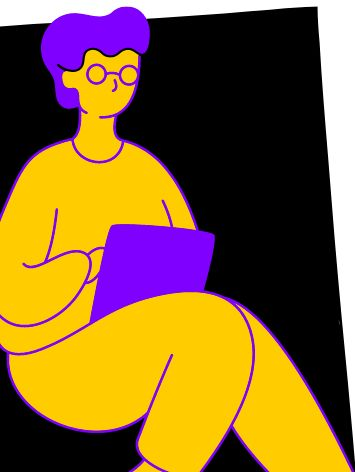
- O jogo tem a participação de dois jogadores e cada jogador tem direito a 3 jogadas.
- A cada jogada o programa sorteia um número de 1 a 6 para cada um dos jogadores. A pontuação do jogador é acumulativa, de acordo com o número sorteado para ele em cada jogada
- No final, o ganhador será o jogador com maior número de pontos na soma da pontuação das 3 jogadas.



EXEMPLOS DE ENTRADA	EXEMPLOS DE SAÍDA
Primeira Jogada Numero sorteado para o Jogador 1: 3 Numero sorteado para o Jogador 2: 1 Segunda Jogada Numero sorteado para o Jogador 1: 4 Numero sorteado para o Jogador 2: 2 Terceira Jogada Numero sorteado para o Jogador 1: 1 Numero sorteado para o Jogador 2: 3	O vencedor foi o Jogador 1 com o total de 8 pontos

EXERCÍCIO 10

Você está responsável por criar o sorteador de números da Mega-Sena! Faça um programa que gere 6 números aleatórios de 01 a 60 e apresente os números sorteados!



ENTRADA	SAÍDA
(esse exercício não possui entrada)	Números sorteados : 20 14 52 3 36 8