

# Oficina de Python e Programação Competitiva

Prof. Marcelo de Souza

Programa de Extensão “Estudos em Engenharia de Software”  
Universidade do Estado de Santa Catarina – UDESC Alto Vale



## Cronograma

- 22/11: Python e programação competitiva
- 29/11: Python e programação competitiva
- 06/12: Desafios de programação

## Metodologia

- Para cada conceito de programação (condicionais, laços de repetição, ...):
  - Problemas práticos de programação competitiva: 1 resolvido + 1 a resolver.

## Ferramentas

- Thonny (e Python!)



# Programação competitiva

*Um esporte mental onde os participantes devem criar programas de acordo com uma especificação, ou seja, programas que resolvam problemas computacionais.*



acm International Collegiate Programming Contest



event sponsor

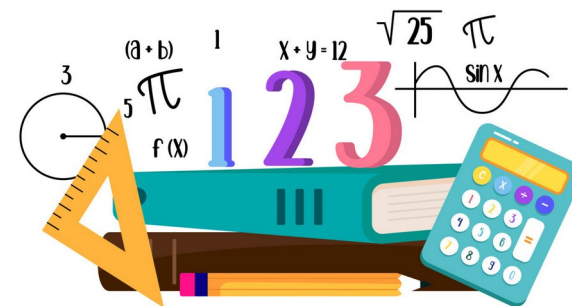




# Problema #1

## Exponenciação

Seu programa deve fazer a leitura de dois números inteiros A e B e apresentar o resultado de  $A^B$ .





# Problema #1

## Exponenciação

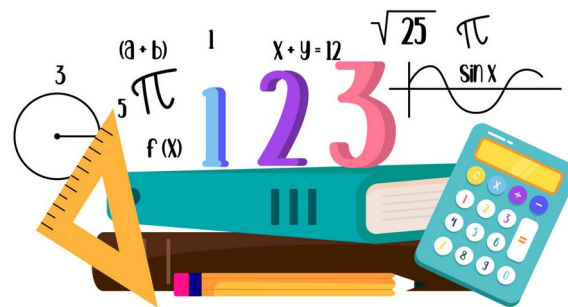
Seu programa deve fazer a leitura de dois números inteiros  $A$  e  $B$  e apresentar o resultado de  $A^B$ . Na entrada, são fornecidos 3 pares de valores  $A$  e  $B$  (cada valor em uma linha). Como saída, deve ser apresentado o resultado de cada uma das 3 operações, um por linha.

## Exemplo de entrada:

```
4
3
2
5
6
2
```

## Saída esperada:

```
64
32
36
```





# Problema #2

## Trapézios

A área de um trapézio é calculada por  $A = (B + b) \times h / 2$ , onde  $B$  é a base maior,  $b$  é a base menor e  $h$  é a altura. A entrada consiste nos valores de  $B$ ,  $b$  e  $h$  de 2 trapézios, todos números naturais e um valor por linha, nessa ordem (total de 6 valores). Seu programa deve calcular a área dos 2 trapézios e apresentar o somatório das áreas, ou seja, um único valor como resultado.

## Exemplo de entrada:

```
1
5
6
7
12
3
```

## Saída esperada:

```
46.5
```





# Problema #3

## Par ou ímpar

Joey e Monica estão em uma disputa de par ou ímpar. Joey aposta no par e Monica aposta no ímpar. Eles desejam jogar  $N$  vezes, e aquele que vencer mais confrontos será o vencedor da disputa. Escreva um programa que auxilie a computar o número de vitórias de cada um e informe quem é o vencedor (ou se eles empataram). Para isso, leia a quantidade  $N$  de disputas, seguida de  $N$  linhas contendo os números apostados por Joey e Monica (na mesma linha). Ao final, informe o vencedor (JOEY ou MONICA) ou então EMPATE.

## Exemplo de entrada:

```
4
5 3
1 4
1 2
3 0
```

## Saída esperada:

```
MONICA
```





# Problema #4

## Andando no tempo

– Maratona de Programação, SBC/ACM, 2016

Imagine que você tenha uma máquina do tempo que pode ser usada no máximo três vezes, e a cada uso da máquina você pode escolher voltar para o passado ou ir para o futuro. A máquina possui três créditos fixos; cada crédito representa uma certa quantidade de anos, e pode ser usado para ir essa quantidade de anos para o passado ou para o futuro. Você pode fazer uma, duas ou três viagens, e cada um desses três créditos pode ser usado uma vez apenas. Por exemplo, se os créditos forem 5, 12 e 9, você poderia decidir fazer duas viagens: ir 5 anos para o futuro e, depois, voltar 9 anos para o passado. Dessa forma, você terminaria 4 anos no passado, em 2017 (considerando o ano atual como 2023). Também poderia fazer três viagens, todas indo para o futuro, usando os créditos em qualquer ordem, terminando em 2049 (considerando o ano atual como 2023).

Neste problema, dados os valores dos três créditos da máquina, seu programa deve dizer se é ou não possível viajar no tempo e voltar para o presente, fazendo pelo menos uma viagem e, no máximo, três viagens; sempre usando cada um dos três créditos no máximo uma vez.

A entrada consiste de uma linha contendo os valores dos três créditos A, B e C. Seu programa deve imprimir uma linha contendo o caractere “S” se é possível viajar e voltar para o presente, ou “N” caso contrário.

### Exemplo de entrada:

3 4 5

### Saída esperada:

N







- SPOJ (DIVSUM)

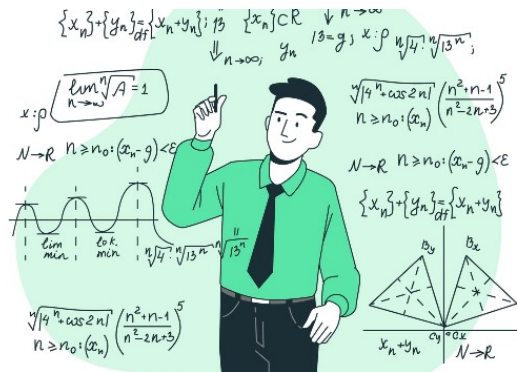
Dado um número natural  $n$  (entre 1 e 500 000), exiba o somatório de todos os seus divisores. Por exemplo, o número 20 tem cinco divisores: 1, 2, 4, 5, 10 e o somatório deles é  $1 + 2 + 4 + 5 + 10 = 22$ . A entrada inicia com o número de casos de teste, seguido de uma linha para cada caso de teste contendo o valor de  $n$ . A saída deve conter uma linha para cada caso de teste, contendo seu resultado.

Exemplo de entrada:

3  
2  
10  
20

Saída esperada:

1  
8  
22





# Problema #6

## Soma no intervalo

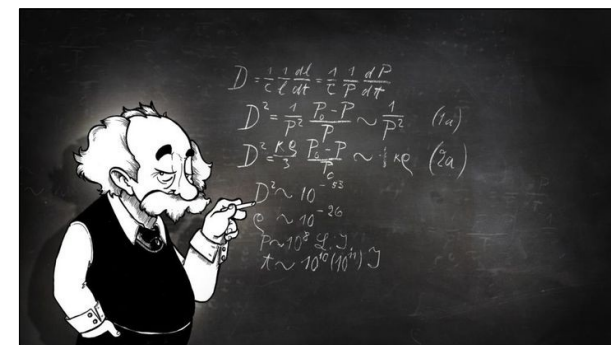
Primeiro, é fornecido o número de casos de teste  $n$ . Depois, são fornecidos  $n$  pares de valores  $x$  e  $y$ , um par por linha, tal que  $x < y$ . Seu programa deverá apresentar a soma de todos os números inteiros entre  $x$  e  $y$  (inclusive). Por exemplo, dado o par 4 e 7, seu programa deve exibir o resultado 22 (que corresponde a  $4 + 5 + 6 + 7$ ).

## Exemplo de entrada:

```
4
2 8
-3 5
42 68
-20 12
```

## Saída esperada:

```
35
9
1485
-132
```





# Problema #7

Mais pontos (e maior salário) que a média

O *Denver Nuggets*, time de basquete da NBA, conseguiu novos patrocínios, o que permite aumentar o salário de alguns dos seus jogadores. No impasse de decidir quais atletas terão o aumento, o estagiário propôs uma ideia: aumentar o salário dos jogadores cuja pontuação na última temporada tenha sido maior que a média de pontuação do time. Por exemplo, se os jogadores do time tiveram uma média de 430 pontos na temporada, todos os jogadores com pontuação superior a 430 receberão aumento. A entrada consiste em uma lista com a pontuação de cada jogador na temporada. A saída esperada é o número de jogadores que receberão aumento de salário.

Exemplo de entrada:

[240, 360, 433, 212, 340]

Saída esperada:

3





# Problema #8

## Can you search?

– SPOJ (AGPC01F)

Shimlin ama brincar com vetores. Ela pediu ao seu amigo fantasma para propor um jogo com vetores. Após pensar um pouco, o fantasma teve uma ideia. O fantasma dirá um número  $b$  e Shimlin terá que buscar o menor número entre os  $b$  primeiros elementos de um dado vetor. O fantasma deu a você a tarefa de encontrar a resposta correta, para que ele possa comparar com a resposta de Shimlin. A entrada consiste em uma linha contendo um vetor de números (de tamanho variável) e uma segunda linha contendo o valor de  $b$ . A saída esperada consiste no menor valor do intervalo correspondente.

## Exemplo de entrada:

```
[4, 7, 2, 5, 3, 0, 6, -8, 4]
```

```
5
```

## Saída esperada:

```
2
```





# Problema #9

## Combiner

– beecrowd (1238)

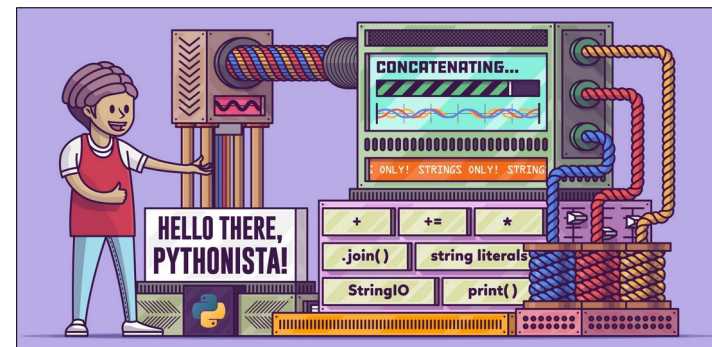
Implemente um programa que leia duas strings e as combine, alternando as letras, iniciando pela primeira letra da primeira string, seguida da primeira letra da segunda string, então a segunda letra da primeira string, etc. As letras restantes da string mais longa são inseridas no final da string combinada. A entrada consiste no número de casos de teste, e uma linha para cada caso de teste, contendo as duas strings. A string combinada para cada caso de teste é apresentada em uma nova linha na saída.

## Exemplo de entrada:

```
2
Tpo oCder
aa bb
```

## Saída esperada:

```
TopCoder
abab
```





# Problema #10

## Dancing sentence

– beecrowd (1234, adaptado)

Uma frase é dita dançante se sua primeira letra estiver em maiúscula, e a caixa de cada letra subsequente é oposta à caixa da letra anterior a ela. Espaços devem ser ignorados ao determinar a caixa das letras. Por exemplo, “A b Cd” é uma frase dançante, pois a primeira letra é maiúscula (“A”), a próxima letra é minúscula (“b”), a seguinte é maiúscula (“C”), e a próxima é minúscula (“d”). Escreva um programa que leia uma frase na entrada (uma linha) e diga se é uma frase dançante (SIM ou NÃO).

## Exemplo de entrada:

Um ExEmPlO dE fRaSe DaNçAnTe

## Saída esperada:

SIM

