
CODE
LAB TEEN
MARATONA
OBI
NÍVEL JÚNIOR

BEM - VINDOS!

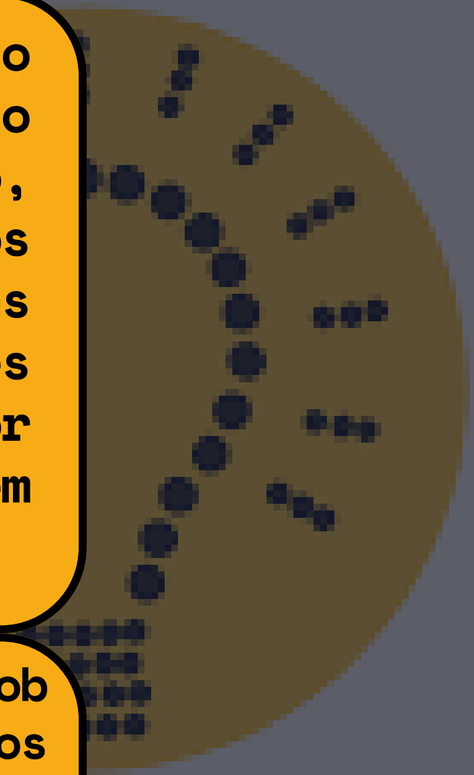
AGENDA

Na Aula de hoje iremos aplicar os conhecimentos de Python em exercícios da modalidade programação da OBI (Olímpiada Brasileira de Informática)! Vamos lá?

JOGO DE PAR OU ÍMPAR

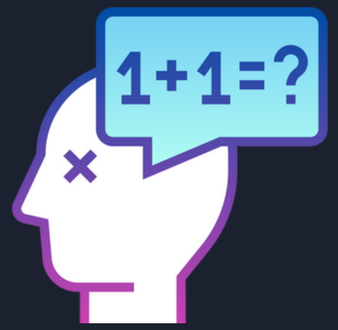
Dois amigos, Alice e Bob, estão jogando um jogo muito simples, em que um deles grita ou "par" ou "ímpar" e o outro imediatamente responde ao contrário, respectivamente "ímpar" ou "par". Em seguida, ambos exibem ao mesmo tempo uma mão cada um, em que alguns dedos estão estendidos e outros dobrados. Então eles contam o número total de dedos estendidos. Se a soma for par, quem gritou "par" ganha. Se a soma for ímpar, quem gritou "ímpar" ganha.

Por exemplo, suponhamos que a Alice gritou "par" e o Bob respondeu "ímpar". Em seguida, Alice não deixou nenhum dos seus dedos estendidos, ao passo que Bob deixou três dedos estendidos. A soma então é três, que é ímpar, portanto Bob ganhou. Seu programa deve determinar quem ganhou, tendo a informação de quem gritou par e o número de dedos estendidos de cada um.



CODE LAB TEEN

JOGO DE PAR OU ÍMPAR



Entrada

A entrada contém três linhas, cada uma com um número inteiro, P, D_1 e D_2, nesta ordem. Se P = 0 então Alice gritou "par", ao passo que se P=1 então Bob gritou "par". Os números D_1 e D_2 indicam, respectivamente, o número de dedos estendidos da Alice e do Bob.

Saída

Seu programa deverá imprimir uma única linha, contendo um único número inteiro, que deve ser 0 se Alice foi a ganhadora, ou 1 se Bob foi o ganhador.

EXEMPLOS

Entrada	Entrada	Entrada
0	1	0
0	0	1
3	3	5
Saída	**Saída**	**Saída**
1	0	0

CODE LAB TEEN

JOGO DE PAR OU ÍMPAR

Resolução:

Primeiro:
Precisamos entender o que estamos recebendo de entrada, no caso: P, D1 e D2.

- ‘P’ representa quem escolheu par;
- D1 número de dados da primeira mão;
- D2 número de dedos da segunda mão;

Assim, elencamos também três variáveis para essas entradas.

Código

```
P = int(input())
D1 = int(input())
D2 = int(input())
```

CODE LAB TEEN

PROVA 2016 fase 1 - programação nível júnior

JOGO DE PAR OU ÍMPAR

Resolução:

Vamos adicionar uma variável que realiza a operação mod (%), que retorna o resto da divisão entre dois números. Essa operação é especialmente útil para verificar se um número é par ou ímpar, pois ao dividir um número por 2, o resto será 0 se for par e 1 se for ímpar.

Código

```
P = int(input())
D1 = int(input())
D2 = int(input())

resultado = (D1 + D2) % 2
```

CODE LAB TEEN

PROVA 2016 fase 1 - programação nível júnior

JOGO DE PAR OU ÍMPAR

Resolução:

Agora, precisamos fazer as verificações de pariedade e de vitória.
Para isso podemos fazer a seguinte solução:

Abrangendo o resultado se Bob ou Alice acertou.

Código

```
P = int(input())
D1 = int(input())
D2 = int(input())

resultado = (D1 + D2) % 2

if resultado == P:
    print(0)  # Alice acertou
else:
    print(1)  # Bob acertou
```

CODE LAB TEEN

PROVA 2016 fase 1 - programação nível júnior

PLANTAÇÃO DE MORANGOS

Os administradores da Fazenda Fartura planejam criar uma nova plantação de morangos, no formato retangular. Eles têm vários locais possíveis para a nova plantação, com diferentes dimensões de comprimento e largura. Para os administradores, o melhor local é aquele que tem a maior área. Eles gostariam de ter um programa de computador que, dadas as dimensões de dois locais, determina o que tem maior área. Você pode ajudá-los?

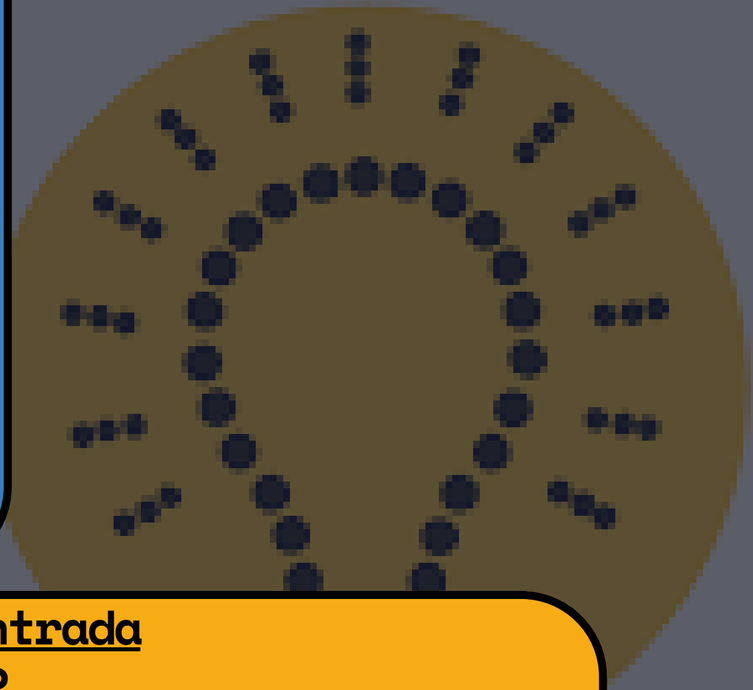
PLANTAÇÃO DE MORANGOS

Entrada

A entrada contém quatro linhas, cada uma contendo um número inteiro. As duas primeiras linhas indicam as dimensões (comprimento e largura) de um dos possíveis locais. As duas últimas linhas indicam as dimensões (comprimento e largura) de um outro possível local para a plantação de morangos. As dimensões são dadas em metros.

Saída

Seu programa deve escrever uma linha contendo um único inteiro, a área, em metros quadrados, do melhor local para a plantação, entre os dois locais dados na entrada.



EXEMPLOS	Entrada	Entrada
	30	12
	8	38
	11	5
	56	20
	Saída	Saída
	616	456

PLANTAÇÃO DE MORANGOS

Resolução:

Relembrando os conceitos de matemática, qual é a fórmula da área de um retângulo?

CODE LAB TEEN

PROVA 2016 fase 1 - programação nível júnior

PLANTAÇÃO DE MORANGOS

Resolução:

Relembrando os conceitos de matemática, qual é a fórmula da área de um retângulo?

Área do retângulo é dado pela multiplicação da sua largura pelo comprimento.

A = L X C

PLANTAÇÃO DE MORANGOS

Resolução:

Sabendo qual a fórmula da Área de uma retângulo, podemos aplicar como função no nosso código! Pois vamos utilizar mais de uma vez dentro do código.

Código

```python
def areaRetangulo(largura, comprimento):
    resultado = largura * comprimento
    return resultado
```

CODE LAB TEEN

PLANTAÇÃO DE MORANGOS

Resolução:

Agora, podemos fazer os Inputs necessários. No enunciado dizia que teríamos 4 inputs (os dois primeiros sendo da comprimento e largura da primeira área, os dois últimos da segunda área).

- C1, L1 para área1
- C2, L2 para área2

Código

```
def areaRetangulo(largura, comprimento):
    resultado = largura * comprimento
    return resultado

# C = comprimento e L = largura
C1 = int(input())
L1 = int(input())
C2 = int(input())
L2 = int(input())

area1 = areaRetangulo(L1, C1)
area2 = areaRetangulo(L2, C2)
```

CODE LAB TEEN

PROVA 2016 fase 1 - programação nível júnior

PLANTAÇÃO DE MORANGOS

Resolução:

Por fim, fazemos uma verificação simples para apresentar a maior área.

CURIOSIDADE: É possível fazer todo esse código com apenas uma linha!

```
print(max(int(input()) * int(input()), int(input()) * int(input())))
```

Area1 Area2

Código

```
def areaRetangulo(largura, comprimento):
    resultado = largura * comprimento
    return resultado

# C = comprimento e L = largura
C1 = int(input())
L1 = int(input())
C2 = int(input())
L2 = int(input())

area1 = areaRetangulo(L1, C1)
area2 = areaRetangulo(L2, C2)
if area1 > area2:
    print(area1)
else:
    print(area2)
```

CODE LAB TEEN

PROVA 2016 fase 1 - programação nível júnior

LÂMPADAS

Você está de volta em seu hotel na Tailândia depois de um dia de mergulhos. O seu quarto tem duas lâmpadas. Vamos chamá-las de A e B. No hotel há dois interruptores, que chamaremos de I_1 e I_2. Ao apertar I_1, a lâmpada A troca de estado, ou seja, acende se estiver apagada e apaga se estiver acesa. Se apertar I_2, ambas as lâmpadas A e B trocam de estado. As lâmpadas inicialmente estão ambas apagadas. Seu amigo resolveu bolar um desafio para você. Ele irá apertar os interruptores em uma certa sequência, e gostaria que você respondesse o estado final das lâmpadas A e B.

CODE LAB TEEN

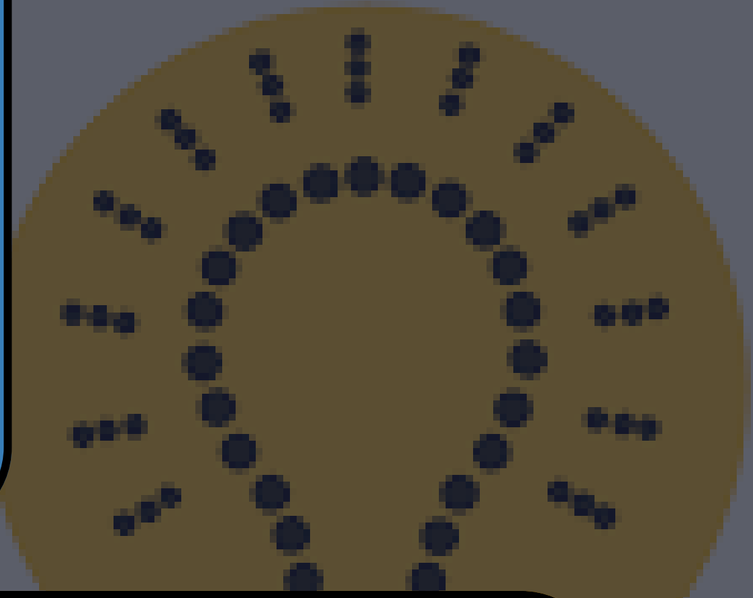
LÂMPADAS

Entrada

A primeira linha contém um número N que representa quantas vezes seu amigo irá apertar algum interruptor. Na linha seguinte seguirão N números, que pode ser 1, se o interruptor I_1 foi apertado, ou 2, se o interruptor I_2 foi apertado.

Saída

Seu programa deve imprimir dois valores, em linhas separadas. Na primeira linha, imprima 1 se a lâmpada A estiver acesa no final das operações e 0 caso contrário. Na segunda linha, imprima 1 se a lâmpada B estiver acesa no final das operações e 0 caso contrário.



EXEMPLOS

Entrada	Entrada
3	4
1 2 2	2 1 2 2
Saída	**Saída**
1	0
0	1

CODE LAB TEEN

LÂMPADAS

Resolução:

Para resolvermos este problema, precisamos lembrar que ocorre mudanças de estado das lâmpadas A e B. Inicialmente vamos colocar A e B com valor 0 (apagado). N recebe a quantidade de vezes que os interruptores I serão acionados

Código

```
A = 0
B = 0
N = int(input())  # quantidade de vezes que será ligado o interruptor
```

CODE LAB TEEN

PROVA 2016 fase 1 - programação nível júnior

LÂMPADAS

Resolução:

Como estamos lidando com uma entrada de dados no formato de uma sequência de números separados por espaços (ex: 1 2 2), precisamos processá-la corretamente para que cada número seja tratado individualmente. Se utilizássemos input() diretamente, a entrada "1 2 2" seria lida como uma única string. Para evitar isso, utilizamos o método .split(), que separa os valores sempre que encontra um espaço em branco, transformando a entrada em uma lista de strings individuais. Isso nos permite acessar cada número separadamente e utilizá-los no programa de forma adequada.
4o

Código

LÂMPADAS

Resolução:

Para percorrermos cada posição do vetor, utilizamos a estrutura de loop for i in range(N), onde N representa a quantidade de vezes que os interruptores foram pressionados.

Dentro do loop, acessamos cada elemento do vetor sequencias, realizando a conversão dos valores quando necessário, para que possamos realizar as verificações e alternações (toggles) nos interruptores corretamente.

Código

```
A = 0
B = 0
N = int(input()) #quantidade de vezes que será ligado o interruptor
sequencias = input().split()

for i in range(N):
    posicao = int(sequencias[i]) # convete para inteiro
```

CODE LAB TEEN

LÂMPADAS

Resolução:

Verificamos a cada posição se o valor 1 (que corresponde ao Interruptor I1) foi pressionado. Caso tenha sido, fazemos a mudança de estado de A para 1. Do contrário, ambos A e B mudam seu estado

Código

```
A = 0
B = 0
N = int(input()) #quantidade de vezes que será ligado o interruptor
sequencias = input().split()

for i in range(N):
    posicao = int(sequencias[i]) # convete para inteiro
    if posicao == 1:
        A = 1 - A # realiza o toggle (mudança de estado)
    else:
        A = 1 - A
        B = 1 - B
print(A)
print(B)
```

CODE LAB TEEN

LÂMPADAS

Resolução:

Uma outra solução para este problema utilizando mais métodos para simplificar o código:

O método map() mapeia e cada valor recebido para inteiro.

^= operação de toggle (alternar).

Código

```
A = B = 0
N = int(input())
sequencias = map(int, input().split())
for comando in sequencias:
    if comando == 1:
        A ^= 1  # Alterna A (toggle)
    else:
        A ^= 1  # Alterna A
        B ^= 1  # Alterna B

print(A)
print(B)
```

FILA

Com a proximidade da Copa do Mundo, o fluxo de pessoas nas filas para compra de ingressos aumentou consideravelmente. Como as filas estão cada vez maiores, pessoas menos pacientes tendem a desistir da compra de ingressos e acabam deixando as filas, liberando assim vaga para outras pessoas. Quando uma pessoa deixa a fila, todas as pessoas que estavam atrás dela dão um passo a frente, sendo assim nunca existe um espaço vago entre duas pessoas. A fila inicialmente contém N pessoas, cada uma com um identificador diferente.
Joãozinho sabe o estado inicial dela e os identificadores em ordem das pessoas que deixaram a fila. Sabendo que após o estado inicial nenhuma pessoa entrou mais na fila, Joãozinho deseja saber o estado final da fila.

CODE LAB TEEN

FILA

Entrada

A primeira linha contém um inteiro N representando a quantidade de pessoas inicialmente na fila. A segunda linha contém N inteiros representando os identificadores das pessoas na fila. O primeiro identificador corresponde ao identificador da primeira pessoa na fila. É garantido que duas pessoas diferentes não possuem o mesmo identificador. A terceira linha contém um inteiro M representando a quantidade de pessoas que deixaram a fila. A quarta linha contém M inteiros representando os identificadores das pessoas que deixaram a fila, na ordem em que elas saíram. É garantido que um mesmo identificador não aparece duas vezes nessa lista.

Saída

Seu programa deve imprimir uma linha contedo N-M inteiros com os identificadores das pessoas que permaneceram na fila, em ordem de chegada.

EXEMPLOS

Entrada	Saída
8 5 100 9 81 70 33 2 1000 3 9 33 5	100 81 70 2 1000

FILA

Resolução:

Para resolver esse exercício podemos tratar todos os dados como String, sem a necessidade de transformamos para Int.

Por estarmos lidando com um grande volume de input, utilizamos o método set(). para termos uma unicidade, eficiência, além de oferecer operações de conjutos (união, diferença e intersecção), o qual estaremos implicitamente utilizando.

Código

```
N = int(input())
fila = input().split()
M = int(input())
saidas = set(input().split())
```

FILA

Resolução:

Após montarmos as variáveis e arrays, criamos um terceiro array. Este recebe a nova fila, no qual recebe apenas os identificadores que não estão no array ‘saidas’.
Essa prática se chama compressão de lista.

Printamos usando o join para saída ficar padronizada com o pedido no enunciado.

Código

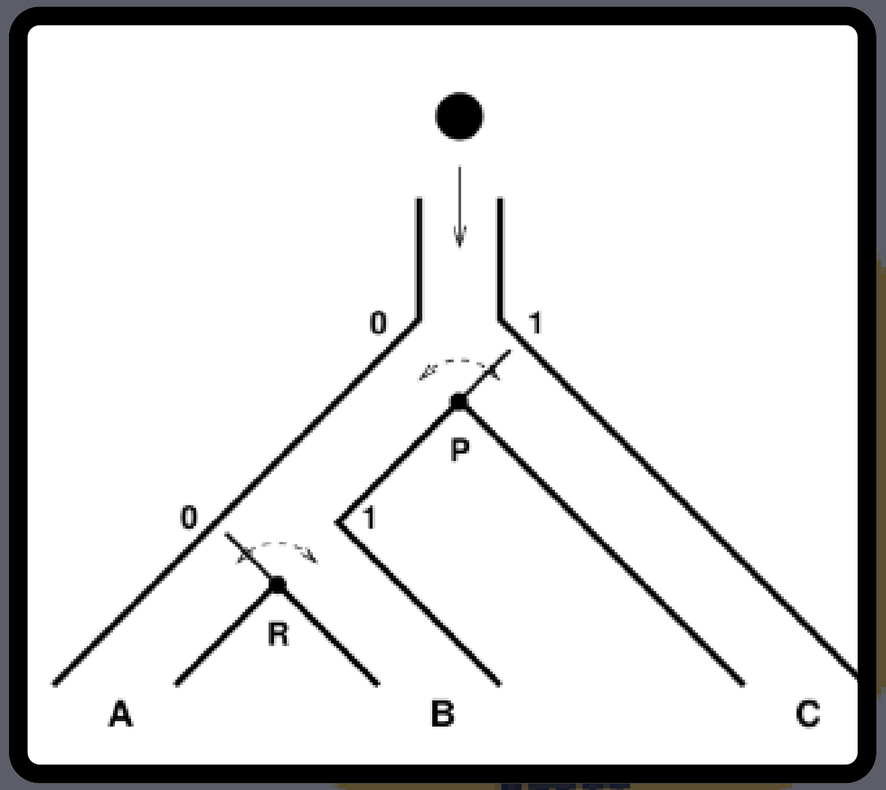
```
N = int(input())
fila = input().split()
M = int(input())
saidas = set(input().split())

fila_final = [p for p in fila if p not in saidas]

print(" ".join(fila_final))
```

FLÍPER

Flíper é um tipo de jogo onde uma bolinha de metal cai por um labirinto de caminhos até chegar na parte de baixo do labirinto. A quantidade de pontos que o jogador ganha depende do caminho que a bolinha seguir. O jogador pode controlar o percurso da bolinha mudando a posição de algumas portinhas do labirinto. Cada portinha pode estar na posição 0, que significa virada para a esquerda, ou na posição 1 que quer dizer virada para a direita. Considere o flíper da figura abaixo, que tem duas portinhas. A portinha P está na posição 1 e a portinha R, na posição 0. Desse jeito, a bolinha vai cair pelo caminho B.



Você deve escrever um programa que, dadas as posições das portinhas P e R, neste flíper da figura, diga por qual dos três caminhos, A, B ou C, a bolinha vai cair!

CODE LAB TEEN

PROVA 2014 fase 1 - programação nível júnior

FLÍPER

Entrada

A entrada é composta por apenas uma linha contendo dois números P e R, indicando as posições das duas portinhas do flíper da figura.

Saída

A saída do seu programa deve ser também apenas uma linha, contendo uma letra maiúscula que indica o caminho por onde a bolinha vai cair: "A", "B" ou "C".

Restrições

- O número P pode ser 0 ou 1. O número R pode ser 0 ou 1.

EXEMPLOS

Entrada
```
1 0
```
Saída
```
B
```

Entrada
```
0 0
```
Saída
```
C
```

FLÍPER

Resolução:

Definimos uma função que recebe os parâmetros P e R, chamada: ‘saida’
Dentro dela fazemos a verificação em qual caminho a bola irá cair.

Como nossa entrada é um vetor, tratamos ela como um array e fazemos a transformação dos dados de entrada para inteiros.
Passamos na chamada da função as posições de P e R.

Código

```
def saida(P, R):
    if P == 0 and R == 0:
        print("C")
    elif P == 1 and R == 0:
        print("B")
    else:
        print("A")

entrada = list(map(int,input().split()))
saida(entrada[0], entrada[1])
```

GANGORRA

Joãozinho acaba de mudar de escola e a primeira coisa que percebeu na nova escola é que a gangorra do parquinho não é simétrica, uma das extremidades é mais longa que a outra. Após brincar algumas vezes com um amigo de mesmo peso, ele percebeu que quando está em uma extremidade, a gangorra se desequilibra para o lado dele (ou seja, ele fica na parte de baixo, e o amigo na parte de cima), mas quando eles trocam de lado, a gangorra se desequilibra para o lado do amigo. Sem entender a situação, Joãozinho pediu ajuda a outro amigo de outra série, que explicou que o comprimento do lado interfere no equilíbrio da gangorra, pois a gangorra estará equilibrada quando

$$P_1 * C_1 = P_2 * C_2$$

onde P_1 e P_2 são os pesos da criança no lado esquerdo e direito, respectivamente, e C_1 e C_2 são os comprimentos da gangorra do lado esquerdo e direito, respectivamente. Com a equação, Joãozinho já consegue dizer se a gangorra está equilibrada ou não mas, além disso, ele quer saber para qual lado a gangorra descerá caso esteja desequilibrada.

CODE LAB TEEN

GANGORRA

Entrada

A primeira e única linha da entrada contém 4 inteiros, P_1, C_1, P_2 e C_2, nesta ordem.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro. Se a gangorra estiver equilibrada, imprima "0". Se ela estiver desequilibrada de modo que a criança esquerda esteja na parte de baixo, imprima "-1", senão, imprima "1".

Restrições

- O número P pode ser 0 ou 1. O número R pode ser 0 ou 1.

EXEMPLOS

Entrada
30 100 60 50
Saída
0

Entrada
40 40 38 60
Saída
1

GANGORRA

Resolução:

Nesta questão faremos uma abordagem direto na main. Criamos as variáveis P1,C1,P2 e C2 e utilizamos da flexibilidade do Python para adicionar cada um dos valores que vieram como vetor em uma variável, usando o map + split.

Para fazer as verificações do equilíbrio da gangorra, fizemos os cálculos direto no if.

Código

```python
P1, C1, P2, C2 = map(int, input().split())

if P1 * C1 == P2 * C2:
    print("0")
elif P1 * C1 > P2 * C2:
    print("-1")
else:
    print("1")
```

CODE LAB TEEN

PROVA 2014 fase 1 - programação nível júnior

OBRIGADO!

Contem para gente o que você achou da aula de hoje:

https://forms.gle/Q1BYFnKxjyKuCC647

CODELAB TEEN