



2ª AVALIAÇÃO

TAREFA:

- Implementar uma solução para cada um dos problemas apresentados a seguir utilizando a linguagem de programação Python 3.

CRITÉRIOS DE AVALIAÇÃO:

- Os códigos apresentados serão avaliados segundo os seguintes critérios:

| ITEM | PESO | CRITÉRIO | DESCRIÇÃO |
|------|------|---------------|---|
| 01 | 40% | Correção | Se o programa realiza a tarefa solicitada |
| 02 | 25% | Solução | Qualidade da solução adotada para o problema |
| 03 | 20% | Implementação | Se o código fonte observa o estilo de codificação Python definida pelo PEP8 |
| 04 | 15% | Formato E/S | Se o programa observa estritamente os formatos da entrada/saída especificados |

- Cada requisito/funcionalidade será avaliado individualmente e receberá uma das seguintes notas:

| NOTA | GRAU DE ATENDIMENTO A CADA CRITÉRIO |
|-------|-------------------------------------|
| 0 | Não atende |
| 1 a 4 | Atende parcialmente |
| 5 | Atende completamente |

- A nota final será determinada pela média ponderada das notas de cada critério.

ORIENTAÇÕES GERAIS:

- A implementação pode ser feita em grupos de ATÉ 2 (dois) alunos.
- A implementação deve **obedecer rigorosamente** os formatos de entrada/saída especificados.
- No início do código fonte **deve haver** um comentário informando o nome dos autores, sob pena de nulidade da questão.
- Os programas devem ler/escrever na entrada/saída padrão.
- Os autores de cada implementação poderão ser chamados para explicar o código entregue.

ENTREGA:

- Ao finalizar o projeto, remeter o **link do repositório GitHub** para o e-mail: eyder@phb.uespi.br com o seguinte assunto: "PROG1 – AVAL2: ", juntamente com os nomes dos membros da equipe no corpo do e-mail;
- A entrega deverá ser feita até o final do dia 03/02/2023 (sexta-feira).
- Códigos considerados copiados da Internet ou de outras equipes receberão NOTA ZERO, independente de quem seja(m) o(s) autor(es).



UNIVERSIDADE ESTADUAL DO PIAUÍ - UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO I
PROFESSOR: EYDER RIOS

PROBLEMA 1:

Uma sequência de N números inteiros (cada número maior que zero) é considerada M -alternada se a sequência for composta por M segmentos consecutivos: o primeiro segmento com um elemento; o segundo segmento com dois elementos; e assim por diante, onde o M -ésimo segmento com M elementos. Além disso, a sequência deve satisfazer as seguintes condições:

1. os elementos de um determinado segmento devem ser todos ímpares ou todos pares;
2. se os elementos em um segmento são todos pares, então os elementos no segmento seguinte devem ser todos ímpares, e vice-versa.

Escreva um programa que, dado um inteiro $N > 1$ e uma sequência de N inteiros positivos, verifique se a sequência é M -alternada. O programa deve computar o valor para M se a sequência for M -alternada, ou produzir a string "NAO" se a sequência não for M -alternada.

FORMATO DA ENTRADA

A primeira linha informa o valor de N . A linha seguinte contém os N inteiros correspondentes à sequência a ser analisada. O arquivo de entrada contém várias instâncias (casos de teste) que terminam com uma linha contendo apenas um "%" (símbolo de porcentagem).

FORMATO DA SAÍDA

Cada linha de saída deve conter o valor de M ou a palavra "NAO", de acordo com as condições descritas acima. O arquivo de saída deve terminar cada saída correspondente a cada instância de entrada por uma linha com um "%" (símbolo de porcentagem).

EXEMPLO

Exemplo de entrada

```
10
12 3 7 2 10 4 5 13 5 11
%
8
11 2 3 4 5 77 33 44
%
```

Saída correspondente

```
4
%
NAO
%
```



PROBLEMA 2:

A Organização Brasileira dos Índios promoverá um festival indígena, onde várias tribos irão se reunir e fazer demonstrações de cultura, como artesanato, danças, pinturas, comidas e etc.

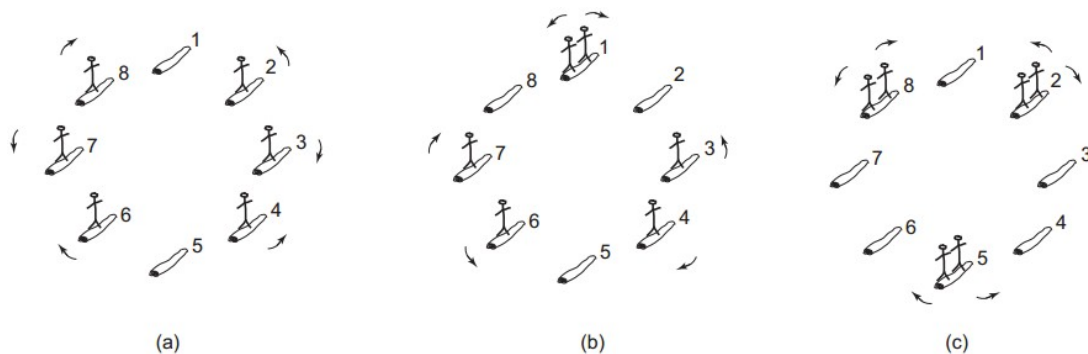
Uma das tribos é a dos Tunak Tunak, que possuem uma apresentação de dança muito peculiar. Nessa dança, existem N toras de madeira encrustadas no chão, dispostas de maneira circular e igualmente espaçadas. Em algumas dessas toras fica um índio, olhando em sentido horário ou anti-horário.

A cada batida do tambor, os índios pulam para a próxima tora (que depende da direção para onde ele está olhando no momento). Durante a dança, porém, algumas coisas podem acontecer:

1. Dois índios que pulam em sentidos opostos caem na mesma tora ao mesmo tempo. Nesse caso, ambos permanecem na tora, mas passam a pular na direção contrária a partir da próxima batida de tambor (isso é, quem estava pulando em sentido horário passa a pular em sentido anti-horário, e vice-versa)
2. Dois índios em toras consecutivas vão pular um em direção ao outro. Nesse caso, os índios simplesmente não pulam (para não causar nenhum acidente), e, assim como no caso anterior, passam a pular no sentido contrário a partir da próxima batida de tambor.
3. Note que se o índio não pula e inverte seu sentido, mas ao mesmo tempo um outro índio cair na mesma tora no sentido contrário, caímos no primeiro caso, e ambos os índios na tora invertem seus sentidos (assim, o índio que estava na tora anteriormente inverte seu sentido novamente).

A dança termina quando as toras ocupadas por cada índio são exatamente as mesmas toras ocupadas no início da dança, não importando qual índio está em cada tora e nem os sentidos para onde eles estão pulando.

A figura abaixo ilustra (a) uma configuração inicial com oito toras e seis índios; (b) a posição dos índios após uma batida de tambor; e (c) a posição dos índios após duas batidas de tambor.



Os índios querem se preparar para a dança e precisam saber quanto tempo ela vai durar. Para isso, você deverá escrever um programa que, dados a quantidade de toras que serão utilizadas, a quantidade de índios e a posição inicial de cada um, calcule quantas batidas de tambor levará para que a dança termine.

FORMATO DA ENTRADA

A primeira linha da entrada possui 2 inteiros: N ($3 \leq N \leq 1.000.000$) e E ($1 \leq E \leq 1000$), que são, respectivamente, a quantidade de toras e a quantidade de índios que irão dançar ($E \leq N$). As próximas E linhas contém, cada uma, a descrição da posição inicial de cada índio. Cada linha possui dois inteiros: V ($1 \leq V \leq N$) e D ($D = 1$ ou $D = -1$) que representam, respectivamente, o número da tora onde o índio inicia e seu sentido inicial (1 se horário, -1 se anti-horário). A numeração das toras obedece o sentido horário. No início da dança uma tora terá, no máximo, um índio.

FORMATO DA SAÍDA

Seu programa deverá exibir um número inteiro representando a quantidade de batidas de tambor necessárias para que a dança acabe.



UNIVERSIDADE ESTADUAL DO PIAUÍ - UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO I
PROFESSOR: EYDER RIOS

Exemplo de entrada #1

6 4
2 1
3 1
5 1
6 1

Saída correspondente #1

3

Exemplo de entrada #2

3 1
2 -1

Saída correspondente #2

3

Exemplo de entrada #3

8 6
2 -1
3 1
4 -1
6 1
7 -1
8 1

Saída correspondente #3

4



UNIVERSIDADE ESTADUAL DO PIAUÍ - UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO I
PROFESSOR: EYDER RIOS

PROBLEMA 3:

Recentemente, uma equipe de exploração descobriu uma tribo perdida na selva amazônica. Os nativos falam uma linguagem muito simples e regular, e para melhorar a comunicação entre nativos e exploradores, você deve escrever um sistema analisador de verbos. Os primeiros contatos com os indígenas já permitiram a dedução das seguintes regras:

1. Todos os verbos são regulares.
2. Todos os verbos terminam com uma consoante seguida pelo sufixo “en”.
3. Existem apenas três tempos verbais: passado, presente e futuro.
4. Para conjugar o verbo, os nativos substituem o sufixo “en” por outro, conforme tabela a seguir:

| Pessoa | Pronome | Presente | Pretérito | Futuro |
|--------|---------|----------|-----------|--------|
| 1ª | Eu | o | ei | ai |
| 2ª | Tu | os | es | ais |
| 3ª | Ele | a | e | i |
| 4ª | Nós | om | em | aem |
| 5ª | Vós | ons | est | aist |
| 6ª | Eles | am | im | aim |

O verbo “**campten**”, por exemplo, possui a seguinte conjugação:

| Pessoa | Pronome | Presente | Pretérito | Futuro |
|--------|---------|----------|-----------|-----------|
| 1ª | Eu | campto | camptei | camptai |
| 2ª | Tu | camptos | camptes | camptais |
| 3ª | Ele | campta | campte | campti |
| 4ª | Nós | camptom | camptem | camptaem |
| 5ª | Vós | camptons | camptest | camptaist |
| 6ª | Eles | camptam | camptim | camptaim |

O programa deve analisar uma palavra, verificar se é verbo ou não e, se for verbo, indicar a pessoa e o tempo do verbo original.

ENTRADA

Uma lista de palavras, uma por linha.

SAÍDA

Uma palavra analisada por linha, seguida do resultado da análise, conforme mostrado nos exemplos.

EXEMPLO

Exemplo de entrada

casos
porre
corraem
picel
oficina
param

Saída correspondente

casos - verbo casen, presente, 2ª pessoa
porre - verbo porren, pretérito, 3ª pessoa
corraem - verbo corren, futuro, 4ª pessoa
picel - não é um tempo verbal
oficina - verbo oficinen, presente, 3ª pessoa
param - verbo paren, presente, 6ª pessoa



PROBLEMA 4:

Considere a seguinte versão do jogo de Dominó:

1. Uma peça de dominó tem dois lados, cada um deles numerado de 0 a 6;
2. O jogo é jogado com 28 peças;
3. Uma peça que tem ambos os lados com o mesmo número é chamada de camburão;
4. Considere uma versão do jogo para dois jogadores: cada jogador recebe um conjunto de peças, onde cada conjunto possui o mesmo número de peças.
5. O jogador que recebeu o camburão de maior valor começa o jogo baixando esta peça, iniciando assim uma configuração (peças na mesa) com dois extremos (pontas).
6. Cada jogador, por sua vez, deve baixar uma de suas peças, sempre em um dos extremos (pontas) da configuração atual, se e somente se, essa peça tiver um de seus lados correspondendo ao número do lado externo do peça que se encontra em um dos extremos (pontas) da configuração que se encontra na mesa. A peça a ser baixada é colocada de forma que os lados com números correspondentes fiquem adjacentes um ao outro. Nenhum tratamento especial é dado a um camburão: ela é posicionada com um de seus lados correspondentes adjacente à peça extrema (ponta).
7. Sempre que um jogador não tiver nenhuma peça que corresponda a um dos extremos, ele (ela) passa sua vez.
8. O vencedor é o jogador que primeiro baixar todas as suas peças.

Suponha que os dois jogadores sejam chamados de AZUL e VERDE. Dados os conjuntos iniciais de peças atribuídos a cada um, seu problema é determinar se:

- a) apenas um deles pode vencer;
- b) ambos podem vencer;
- c) nenhum deles pode vencer o jogo.

Observe que este não é um problema de encontrar estratégias vencedoras. Você apenas tem que descobrir a possibilidade ou não de vitória de cada um dos jogadores.

FORMATO DA ENTRADA

O arquivo de entrada pode conter diversas instâncias (casos de teste) do problema. Cada instância consiste em uma linha com o seguinte formato:

$$N \ A_{11} \ A_{21} \ A_{12} \ A_{22} \ \dots \ A_{N1} \ A_{N2} \ V_{11} \ V_{12} \ \dots \ V_{N1} \ V_{N2}$$

onde:

N número de peças atribuídas inicialmente para cada um dos jogadores AZUL e VERDE;
 $A_{i1} \ A_{i2}$ indicam cada um dos dois lados da i -ésima peça atribuída ao jogador AZUL
 $V_{i1} \ V_{i2}$ indicam cada um dos dois lados da i -ésima peça atribuída ao jogador VERDE

de forma que:

- a) Não há repetição de peças;
- b) Pelo menos um dos jogadores recebe um camburão;
- c) cada A_{i1} , A_{i2} , V_{i1} e V_{i2} é um número inteiro entre 0 e 6, inclusive;
- d) os valores em cada linha podem estar separados por um número arbitrário de espaços;
- e) instâncias consecutivas não são separadas por uma linha em branco;
- f) a última linha contém um 0.



FORMATO DA SAÍDA

Para cada instância do problema, o programa deve gerar uma linha contendo uma das seguintes quatro mensagens:

- Somente o jogador AZUL pode vencer
- Somente o jogador VERDE pode vencer
- Ambos os jogadores podem vencer
- Nenhum jogador pode vencer

EXEMPLO

Exemplo de entrada

```
1 6 6 0 0
2 5 5 3 4 1 5 3 5
2 1 1 2 2 3 3 4 4
1 4 5 4 4
0
```

Saída correspondente

```
Somente o jogador AZUL pode vencer
Ambos os jogadores podem vencer
Nenhum jogador pode vencer
Somente o jogador VERDE pode vencer
```