



## **2ª AVALIAÇÃO**

**25/02/2022**

### **ORIENTAÇÕES GERAIS:**

1. Siga atentamente as especificações da tarefa quanto ao formato da entrada e saída de seu programa.
2. Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
3. Todas as tarefas têm o mesmo valor na correção.
4. As tarefas não estão ordenadas por ordem de dificuldade.
5. Não utilize arquivos para entrada ou saída nos programas. Todos os dados devem ser lidos da entrada padrão (tipicamente o teclado) e escritos na saída padrão (tipicamente a tela).
6. Procure resolver o problema de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.
7. A implementação pode ser feitas em grupos de ATÉ 3 (três) alunos.
8. No início do código fonte **deve haver um comentário informando os nomes dos autores**, sob pena de nulidade da questão.
9. Os autores de cada implementação poderão ser chamados para explicar o código entregue.

### **ENTREGA:**

- Os códigos produzidos devem ser entregues em **um único arquivo compactado** em formato ZIP.
- O arquivo compactado deverá ser enviado para o e-mail: [eyder@phb.uespi.br](mailto:eyder@phb.uespi.br)
- O **assunto do e-mail** deverá iniciar com “PROG1 – AVAL2: ” seguido dos primeiros nomes dos autores.
- **A entrega deverá ser feita até as dia 27/02/2022.**
- Códigos considerados copiados da Internet ou de outras equipes receberão NOTA ZERO, independente de quem seja(m) o(s) autor(es).



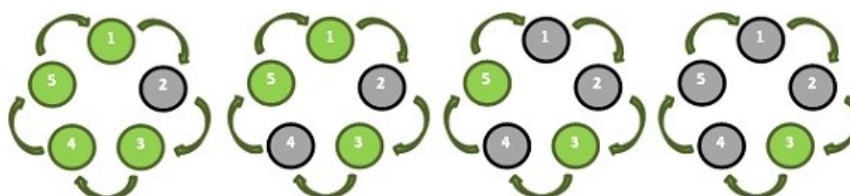
### **PROBLEMA 1:**

Segundo um historiador judeu Hassan que viveu no século I, durante o cerco da cidade de Atlantis, ele e seus 40 companheiros soldados ficaram presos em uma caverna, cuja saída foi bloqueada pelo exército romano. Eles escolheram o suicídio em vez da captura e decidiram que formariam um círculo e começariam a se matar pulando de três em três. Hassan, por sorte ou talvez pela mão de Deus, sobreviveu ao suicídio coletivo e se entregou aos romanos.

### **FORMATO DA ENTRADA**

A primeira linha da entrada contém um número inteiro  $C$  ( $1 \leq C \leq 30$ ) indicando a quantidade de casos de testes da entrada. Caso de teste é composto por um única linha contendo um par de números inteiros positivos  $N$  ( $1 \leq N \leq 10000$ ) e  $K$  ( $1 \leq K \leq 1000$ ). O número  $N$  representa a quantidade de soldados no círculo, numerados de 1 a  $N$ . O número  $K$  representa o tamanho do salto entre dois homens no círculo.

A figura a seguir ilustra um exemplo com  $N=5$  soldados e salto  $K=2$ . Neste exemplo o soldado sobrevivente é o de número 3.



Sua tarefa é escrever um programa que para cada caso de teste, exiba qual o número do soldado sobrevivente.

### **FORMATO DA SAÍDA**

Para cada caso de teste deverá ser exibida uma única linha contendo o número do soldado sobrevivente.

#### **Exemplo de entrada**

```
3
5 2
6 3
1234 233
```

#### **Saída correspondente**

```
3
1
25
```



## **PROBLEMA 2:**

A luta continua, seja para armazenar números começando com o dígito mais significativo ou com o dígito menos significativo. Às vezes, isso também é chamado de "Guerra Endian". O campo de batalha remonta aos primórdios da Ciência da Computação. Joe Stoy, em seu livro "Semântica Denotacional", conta a seguinte história:

*"A decisão para que lado os dígitos rodam é, claro, matematicamente trivial. De fato, um computador britânico antigo tinha números correndo da direita para a esquerda (porque o ponto em um tubo de osciloscópio vai da esquerda para a direita, mas na lógica serial o menos dígitos significativos são tratados primeiro). Turing costumava mistificar o público em palestras públicas quando, por acaso, ele entrava nesse modo mesmo para aritmética decimal e escrevia coisas como  $73+42=16$ . A próxima versão da máquina era tornada mais convencional simplesmente cruzando os fios de deflexão x: isso, no entanto, preocupou os engenheiros, cujas formas de onda estavam todas invertidas. Esse problema foi resolvido por sua vez, fornecendo uma pequena janela para que os engenheiros (que tendiam a ficar atrás do computador de qualquer maneira) pudessem ver a tela do osciloscópio na parte de trás.*

*[C. Strachey - comunicação privada.]"*

Você fará o papel do público e julgará o valor de verdade das equações de Turing.

### **FORMATO DA ENTRADA**

A entrada contém vários casos de teste. Cada caso de teste especifica em uma única linha uma equação de Turing. Uma equação de Turing tem a forma " $A+B=C$ ", onde A, B, C são números formados pelos dígitos de 0 a 9, inclusive. Cada número terá no máximo 7 dígitos. Isso inclui possíveis zeros à esquerda ou à direita. A equação " $0+0=0$ " finalizará a entrada e também deverá ser processada. As equações não conterão espaços.

### **FORMATO DA SAÍDA**

Para cada caso de teste gere uma linha contendo a letra "V" ou a letra "F", se a equação for verdadeira ou falsa, respectivamente, na interpretação de Turing, ou seja, os números sendo lidos de trás para frente.

#### **Exemplo de entrada**

73+42=16  
5+8=13  
10+20=30  
0001000+000200=00030  
1234+5=1239  
1+0=0  
7000+8000=51  
0+0=0

#### **Saída correspondente**

V  
F  
V  
V  
F  
F  
V  
V



### **PROBLEMA 3:**

Os habitantes de Brazilândia são muito supersticiosos. Uma de suas crenças é que os números das casas em uma rua que têm um dígito repetido trazem azar para os moradores. Portanto, eles nunca viveriam em uma casa com número de rua como 838 ou 1004.

A Rei de Brazilândia mandou construir uma nova avenida à beira-mar e quer atribuir às novas casas apenas números sem dígitos repetidos, para evitar desconforto entre seus súditos. Você foi nomeado por Sua Majestade para escrever um programa que, dados dois inteiros N e M, determine o número máximo de casas que podem receber números de rua entre N e M, inclusive, que não tenham dígitos repetidos.

### **ENTRADA**

Cada caso de teste é descrito em uma única linha. A linha contém dois inteiros N e M, conforme descrito acima ( $1 \leq N \leq M \leq 5000$ ).

### **SAÍDA**

Para cada caso de teste, imprima uma linha com um inteiro representando o número de números de casas de rua entre N e M, inclusive, sem dígitos repetidos.

### **EXEMPLO**

<b>Exemplo de entrada</b>	<b>Saída correspondente</b>
87 104	14
989 1022	0
22 25	3
1234 1234	1



#### **PROBLEMA 4:**

Muitas crianças gostam de decidir todas as disputas através do famoso jogo de Par ou Ímpar. Nesse jogo, um dos participantes escolhe Par e o outro Ímpar. Após a escolha, os dois jogadores mostram, simultaneamente, uma certa quantidade de dedos de uma das mãos. Se a soma dos dedos das mãos dos dois jogadores for par, vence o jogador que escolheu Par inicialmente, caso contrário vence o que escolheu Ímpar.

Dada uma sequência de informações sobre partidas de Par ou Ímpar (nomes dos jogadores e números que os jogadores escolheram), você deve escrever um programa para indicar o vencedor de cada uma das partidas

#### **FORMATO DA ENTRADA**

A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um inteiro  $N$  ( $0 \leq N \leq 1000$ ), que indica o número de partidas de Par ou Ímpar que aconteceram. As duas linhas seguintes contêm cada uma um nome de jogador. Um nome de jogador é uma cadeia de no mínimo um e no máximo dez letras (maiúsculas e minúsculas), sem espaços em branco. As  $N$  linhas seguintes contêm cada uma dois inteiros  $A$  e  $B$  que representam o número de dedos que cada jogador mostrou em cada partida ( $0 \leq A \leq 5$  e  $0 \leq B \leq 5$ ). Em todas as partidas, o primeiro jogador sempre escolhe Par. O final da entrada é indicado por  $N = 0$ .

#### **FORMATO DA SAÍDA**

Para cada conjunto de teste da entrada, seu programa deve produzir a saída da seguinte forma. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste  $n$ ”, onde  $n$  é numerado sequencialmente a partir de 1. As próximas  $N$  linhas devem indicar o nome do vencedor de cada partida. A próxima linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

#### **EXEMPLO**

##### **Exemplo de entrada**

```
3
Pedro
Paulo
2 4
3 5
1 0
2
Claudio
Carlos
1 5
2 3
0
```

##### **Saída correspondente**

```
Teste 1
Pedro
Pedro
Paulo

Teste 2
Claudio
Carlos
```



### **PROBLEMA 5:**

O Campeonato de Surf do Reino de Brazilândia, que será realizado na cidade de Bela Praia (BP), está prestes a começar! Antes de ir para BP, você e seus amigos  $N - 1$  decidiram ir comer uma pizza, para relaxar e se divertir (e, claro, comer!).

Neste momento você está escolhendo a data para o evento. Para garantir que todos possam aproveitar, foi decidido que a reunião será marcada em um dia em que todas as  $N$  pessoas possam comparecer à pizzeria nessa data.

Dada a lista de datas consideradas para o evento e as informações sobre quais pessoas podem comparecer em quais datas, determine se o evento pode acontecer e, se puder, sua data. Se for possível mais de uma data, o evento deve ocorrer o mais cedo possível.

### **FORMATO DA ENTRADA**

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém os inteiros  $N$  e  $D$  ( $1 \leq N, D \leq 50$ ), indicando, respectivamente, o número de pessoas e o número de datas consideradas. As pessoas são numeradas de 1 a  $N$ . As próximas  $D$  linhas descrevem as datas consideradas. Cada linha começa com uma data no formato “dia/mês/ano”. A linha é seguida por  $N$  inteiros  $p_1, p_2, \dots, p_N$ . O inteiro  $p_i$  é 1 se a  $i$ -ésima pessoa puder aparecer na data considerada, ou 0 caso contrário. É garantido que as datas são sempre válidas e não há zeros à esquerda. Uma linha contendo  $N=0$  e  $D=0$  indica o final da entrada.

### **FORMATO DA SAÍDA**

Para cada caso de teste, imprima uma linha com a data no formato “dia/mês/ano”, exatamente como aparece na entrada. Caso não seja possível realizar o evento, imprima “N/P” (sem aspas).

### **EXEMPLO**

#### **Exemplo de entrada**

```
4 4
1/6/2016 0 0 1 0
12/7/2016 1 1 1 0
5/10/2016 1 1 1 1
25/12/2016 0 0 0 0
5 3
20/9/2016 0 1 1 1 1
30/9/2016 1 0 1 1 1
1/10/2016 1 1 0 1 1
0 0
```

#### **Saída correspondente**

```
5/10/2016
N/P
```