

Competidor(a): _____

Número de inscrição: _____ – _____ (*opcional*)



OBI2019

Caderno de Tarefas

Modalidade **Programação • Nível 2 • Fase Local**

30 de maio de 2019

A PROVA TEM DURAÇÃO DE 2 HORAS

Promoção:



Sociedade Brasileira de Computação

Apoio:



Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 7 páginas (não contando a folha de rosto), numeradas de 1 a 7. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa; em particular, seu programa não deve escrever frases como “Digite o dado de entrada:” ou similares.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão necessariamente ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 2 devem ser arquivos com sufixo *.py2*; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py3*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*.
- Na linguagem Java, **não** use o comando *package*, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Para tarefas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada tarefa.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou pen-drive, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
 - em Pascal: *readln, read, writeln, write*;
 - em C: *scanf, getchar, printf, putchar*;
 - em C++: as mesmas de C ou os objetos *cout* e *cin*.
 - em Java: qualquer classe ou função padrão, como por exemplo *Scanner, BufferedReader, BufferedWriter* e *System.out.println*
 - em Python: *read, readline, readlines, input, print, write*
 - em Javascript: *scanf, printf*
- Procure resolver a tarefa 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.

Calçada Imperial

Nome do arquivo: “imperial.x”, onde x deve ser `c`, `cpp`, `pas`, `java`, `js`, `py2` ou `py3`

Na calçada em frente ao Palácio Imperial, não se sabe a razão, existe uma sequência de N números desenhados no chão. A sequência é composta apenas pelos números de 1 a N . Veja um exemplo na coluna (a) da figura ao lado, para $N = 12$.

Ninguém sabe o significado da sequência e, justamente por isso, várias teorias malucas surgiram. Uma delas diz que a sequência representa, na verdade, apenas um valor que estaria relacionado a um grande segredo dos imperadores. Esse valor é a quantidade *máxima* de números da sequência que poderiam ser marcados com um círculo, de modo que a sequência de números marcados não contenha dois números iguais consecutivos e seja composta de no máximo dois números distintos.

A coluna (b) da figura ilustra uma sequência de 4 números marcados que obedece a restrição acima. Você consegue verificar que essa é, de fato, a quantidade máxima possível de números numa sequência marcada?

3	3
2	②
5	5
2	2
10	⑩
4	4
4	4
7	7
12	12
2	②
8	8
10	⑩
(a)	(b)

Neste problema, dada a sequência original de números desenhados no chão da calçada, seu programa deve computar e imprimir a quantidade máxima de números da sequência que poderiam ser marcados com um círculo sem que haja dois números iguais consecutivos na sequência marcada e tal que ela seja composta de no máximo dois números distintos.

Entrada

A primeira linha da entrada contém um inteiro N representando o tamanho da sequência. As N linhas seguintes contêm, cada uma, um inteiro V_i , para $1 \leq i \leq N$, definindo a sequência de números desenhados no chão da calçada imperial.

Saída

Seu programa deve imprimir uma linha contendo um número inteiro representando a quantidade máxima de números da sequência que poderiam ser marcados com um círculo sem que haja dois números iguais consecutivos na sequência marcada e tal que ela seja composta de no máximo dois números distintos.

Restrições

- $1 \leq N \leq 500$
- $1 \leq V_i \leq N$, para $1 \leq i \leq N$

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
1 1	1

Exemplo de entrada 2	Exemplo de saída 2
12 3 2 5 2 10 4 4 7 12 2 8 10	4

A idade de Dona Mônica

Nome do arquivo: “idade.x”, onde x deve ser `c`, `cpp`, `pas`, `java`, `js`, `py2` ou `py3`

Dona Mônica é mãe de três filhos que têm idades diferentes. Ela notou que, neste ano, a soma das idades dos seus três filhos é igual à idade dela. Neste problema, dada a idade de dona Mônica e as idades de dois dos filhos, seu programa deve computar e imprimir a idade do filho mais velho.

Por exemplo, se sabemos que dona Mônica tem 52 anos e as idades conhecidas de dois dos filhos são 14 e 18 anos, então a idade do outro filho, que não era conhecida, tem que ser 20 anos, pois a soma das três idades tem que ser 52. Portanto, a idade do filho mais velho é 20. Em mais um exemplo, se dona Mônica tem 47 anos e as idades de dois dos filhos são 21 e 9 anos, então o outro filho tem que ter 17 anos e, portanto, a idade do filho mais velho é 21.

Entrada

A primeira linha da entrada contém um inteiro M representando a idade de dona Mônica. A segunda linha da entrada contém um inteiro A representando a idade de um dos filhos. A terceira linha da entrada contém um inteiro B representando a idade de outro filho.

Saída

Seu programa deve imprimir uma linha, contendo um número inteiro, representando a idade do filho mais velho de dona Mônica.

Restrições

- $40 \leq M \leq 110$
- $1 \leq A < M$
- $1 \leq B < M$
- $A \neq B$

Exemplos

Exemplo de entrada 1 52 14 18	Exemplo de saída 1 20
Exemplo de entrada 2 47 21 9	Exemplo de saída 2 21

Soma

Nome do arquivo: “soma.x”, onde x deve ser `c`, `cpp`, `pas`, `java`, `js`, `py2` ou `py3`

Temos uma sequência de N quadrados desenhados lado a lado. Cada quadrado possui um número natural anotado dentro dele. Dados a sequência dos N quadrados e um valor K , quantos retângulos distintos existem cuja soma dos números dentro do retângulo é exatamente igual a K ? Por exemplo, a figura mostra uma sequência de $N = 10$ quadrados para a qual existem 5 retângulos cuja soma dos números é igual a $K = 4$.

2	0	1	1	0	0	8	4	1	3
2	0	1	1	0	0	8	4	1	3
2	0	1	1	0	0	8	4	1	3
2	0	1	1	0	0	8	4	1	3
2	0	1	1	0	0	8	4	1	3

Entrada

A primeira linha da entrada contém dois inteiros N e K representando o número de quadrados na sequência e o valor da soma desejada. A segunda linha da entrada contém N números naturais X_i , para $1 \leq i \leq N$, indicando a sequência de números anotados dentro dos quadrados.

Saída

Seu programa deve imprimir uma linha contendo um número inteiro representando quantos retângulos existem na sequência cuja soma é igual a K .

Restrições

- $1 \leq N \leq 500000$ (5×10^5)
- $0 \leq K \leq 10^6$
- $0 \leq X_i \leq 100$ para $1 \leq i \leq N$

Informações sobre a pontuação

- Em um conjunto de casos de teste somando 10 pontos, $N \leq 500$
- Em um conjunto de casos de teste somando 20 pontos, $N \leq 10^4$
- Em um conjunto de casos de teste somando 30 pontos, $K > 0$ e $X_i > 0$ para $1 \leq i \leq N$
- Em um conjunto de casos de teste somando 40 pontos, nenhuma restrição adicional (*note que para esta subtarefa o inteiro da saída pode não caber em 32 bits.*)

Exemplos

<p>Exemplo de entrada 1</p> <p>10 4 2 0 1 1 0 0 8 4 1 3</p>	<p>Exemplo de saída 1</p> <p>5</p>
<p>Exemplo de entrada 2</p> <p>15 0 0 0 0 0 0 5 12 0 1 0 0 0 51 0 0</p>	<p>Exemplo de saída 2</p> <p>25</p>

Chuva

Nome do arquivo: “chuva.x”, onde x deve ser `c`, `cpp`, `pas`, `java`, `js`, `py2` ou `py3`

Está chovendo tanto na Obilândia que começaram a aparecer goteiras dentro da casa do prefeito. Uma dessas goteiras está fazendo escorrer água verticalmente, a partir de um ponto no teto, numa parede onde há várias prateleiras horizontais. Quando a água bate em uma prateleira, ela começa a escorrer horizontalmente para os dois lados, direita e esquerda, até as extremidades da prateleira, quando volta a escorrer verticalmente.

Vamos representar a parede por uma matriz de N linhas e M colunas de caracteres, como mostrado ao lado. As prateleiras serão representadas por “#” e a parede por “.”. Só existem prateleiras nas linhas pares e elas nunca encostam na borda da parede. Há apenas um ponto de vazamento representado pelo caractere “o” na primeira linha.

Para deixar mais rigorosa a forma como a água vai escorrer, seja $c(i, j)$ o caractere na linha i coluna j . Se $c(i, j) = “.”$, então ele deve virar “o” sempre que:

- $c(i - 1, j) = “o”$; ou
- $c(i, j - 1) = “o”$ e $c(i + 1, j - 1) = “#”$; ou
- $c(i, j + 1) = “o”$ e $c(i + 1, j + 1) = “#”$.

Neste problema, dada a matriz representando a parede no início do vazamento, seu programa deve imprimir na saída uma matriz representando a parede usando o caractere “o” exatamente nas posições que serão molhadas pelo vazamento, como ilustrado acima.

Entrada

A primeira linha da entrada contém dois inteiros N e M , respectivamente o número de linhas e colunas da matriz. As N linhas seguintes da entrada contêm, cada uma, uma sequência de M caracteres entre três possíveis: “.”, “#” ou “o”.

Saída

Seu programa deve imprimir N linhas, cada uma contendo uma sequência de M caracteres, representando a matriz da entrada usando o caractere “o” exatamente nas posições que serão molhadas pelo vazamento.

Restrições

- $3 \leq N \leq 500$ e $3 \leq M \leq 500$;
- O número de linhas N é ímpar;
- Há exatamente um caractere “o” na primeira linha;
- As linhas ímpares, a primeira coluna e a última coluna não possuem o caractere “#”.

```
.....o.....
.###...####.#.
.....
.#####.....
.....
.#.####...##.
.....
.....#####.
.....
```

```
.....o.....
.###...o####o#.
.o.....o.....
.o#####o.....
ooo.....o.oooo
o#o####.o.o##o
o.o.ooooo.....
o.o.o####oo.o.
o.o.o.....oo.o.
```

Exemplos

<p>Exemplo de entrada 1</p> <p>9 14</p> <p>.....o.....</p> <p>.###...###.#.</p> <p>.....</p> <p>..#####.....</p> <p>.....</p> <p>.#.####...##.</p> <p>.....</p> <p>.....#####.....</p> <p>.....</p>	<p>Exemplo de saída 1</p> <p>.....000000..</p> <p>.###..o####o#.</p> <p>.00000000..o..</p> <p>.o#####o..o..</p> <p>000.....o.0000</p> <p>o#o####.o.o##o</p> <p>o.o.0000000..o</p> <p>o.o.o####oo..o</p> <p>o.o.o....oo..o</p>
<p>Exemplo de entrada 2</p> <p>7 5</p> <p>.o...</p> <p>.#...</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>..#..</p> <p>.....</p>	<p>Exemplo de saída 2</p> <p>ooo..</p> <p>o#o..</p> <p>o.o..</p> <p>o.o..</p> <p>oooo.</p> <p>oo#o.</p> <p>oo.o.</p>
<p>Exemplo de entrada 3</p> <p>3 3</p> <p>..o</p> <p>.#.</p> <p>...</p>	<p>Exemplo de saída 3</p> <p>..o</p> <p>.#o</p> <p>..o</p>