Competidor(a): _____

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

Este Caderno de Tarefas não pode ser levado para casa após a prova. Após a prova entregue este Caderno de Tarefas para seu professor guardar. Os professores poderão devolver os Cadernos de Tarefas aos competidores após o término do período de aplicação das provas (23 de agosto de 2022).

Olimpíada Brasileira de Informática

OBI2022

Caderno de Tarefas

Modalidade Programação • Nível Júnior • Fase 2

23 de agosto de 2022

A PROVA TEM DURAÇÃO DE 2 HORAS

Promoção:

Sociedade Brasileira de Computação

Apoio:

FUNDAÇÃO BEHRING

Coordenação:

INSTITUTO DE COMPUTAÇÃO UNICAMP

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 3 devem ser arquivos com sufixo *.py*; 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.

Troféu

Nome do arquivo: `trofeu.c`, `trofeu.cpp`, `trofeu.pas`, `trofeu.java`, `trofeu.js` *ou* `trofeu.py`

Cinco alunos e alunas da escola conseguiram classificar-se para a Final da prestigiosa e muito difícil Competição Estadual de Programação, que será realizada no próximo mês.

Independentemente da classificação que os alunos da escola conseguirem na Final, a direção da escola decidiu que vai fazer uma premiação para os seus alunos. Quem conseguir a maior pontuação na Final, entre os alunos da escola, vai receber um troféu. E quem receber a segunda maior pontuação, entre os alunos da escola, vai receber uma placa comemorativa.

O problema é que pode haver alunos com as mesmas pontuações, de forma que dependendo dos resultados muitas combinações de prêmios são possíveis, como por exemplo, entre outros:

- cinco troféus (empate, todos com a mesma pontuação)
- um troféu (maior pontuação) e duas placas (empate na segunda maior pontuação)
- dois troféus (empate na maior pontuação) e duas placas (empate na segunda maior pontuação)

Dadas as pontuações dos cinco alunos e alunas, determine quantos troféus e placas deverão ser entregues.

Entrada

A entrada consiste de cinco linhas, cada uma contendo um inteiro P_i a pontuação de um aluno ou aluna. As pontuações serão dadas em ordem decrescente (ou seja, da maior para a menor pontuação).

Saída

Seu programa deve produzir uma única linha, contendo dois inteiros. O primeiro inteiro deve ser o número de troféus e o segundo inteiro o número de placas comemorativas a serem entregues.

Restrições

- $1 \leq P_i \leq 100$

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**
100 90 80 70 60	1 1

Explicação do exemplo 1: A maior pontuação (100) ganha o troféu. A segunda maior pontuação (90) ganha a placa comemorativa.

Exemplo de entrada 2	**Exemplo de saída 2**
100 100 90 89 77	2 1

Explicação do exemplo 2: Há um empate na maior pontuação (100), portanto os dois ganham troféus. A segunda maior pontuação (90) ganha a placa comemorativa.

Exemplo de entrada 3	**Exemplo de saída 3**
99 99 99 99 99	5 0

Explicação do exemplo 3: Há um empate na maior pontuação (99), portanto os cinco ganham troféus. Não há entrega de placa comemorativa neste caso.

Caminho

Nome do arquivo: `caminho.c`, `caminho.cpp`, `caminho.pas`, `caminho.java`, `caminho.js` *ou* `caminho.py`

A pista de treinos de corridas da Prefeitura tem formato circular, com N postes igualmente espaçados na circunferência da pista, cada poste com exatamente uma lâmpada. Atualmente há lâmpadas de várias potências luminosas nos postes.

Vamos chamar o trecho de pista entre duas lâmpadas adjadentes (ou seja, uma vizinha à outra) de *trecho escuro* se a soma das potências luminosas dessas duas lâmpadas é menor do que 1000.

Para justificar um pedido à Prefeitura para que troquem as lâmpadas, os atletas querem saber qual o maior número de trechos escuros consecutivos (ou seja, um imediatamente em seguida de outro) da pista. Você pode ajudá-los?

Entrada

A primeira linha da entrada contém um inteiro N, o número de postes. Cada uma das N linhas seguintes contém um inteiro P_i, a potência luminosa de uma lâmpada. A posição de cada lâmpada é dada pela ordem da entrada (ou seja, a ordem das lâmpadas ao longo da pista é a ordem dada na entrada).

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o maior número de trechos escuros consecutivos.

Restrições

- $2 \leq N \leq 500\ 000$
- $1 \leq P_i \leq 1\ 000$ para $i \leq i \leq N$

Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 40 pontos, $P_1 + P_N \geq 1\ 000$.

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**
3 499 500 499	3

Explicação do exemplo 1: O maior (e único) caminho de pares de lâmpadas com potência luminosa abaixo do especificado é formado pelos pares (499,500), (500, 499) e (499, 499), portanto a resposta é 3.

Exemplo de entrada 2	**Exemplo de saída 2**
6 900 700 100 900 200 700	1

Explicação do exemplo 2: O único par de lâmpadas com potência luminosa abaixo do especificado é (700,100), portanto a resposta é 1.

Exemplo de entrada 3	**Exemplo de saída 3**
6 500 800 290 700 200 400	4

Explicação do exemplo 3: O maior (e único caminho) com pares de lâmpadas com potência menor do que o especificado é formado pelos pares (290,700), (700,200), (200,400) e (400,500), portanto a resposta é 4.

Exemplo de entrada 4	**Exemplo de saída 4**
2 500 600	0

Explicação do exemplo 4: Não há nenhum par de lâmpadas com potência abaixo do especificado, portanto a resposta é 0.

Exemplo de entrada 5	**Exemplo de saída 5**
2 100 101	2

Explicação do exemplo 5: O maior (e único caminho) com pares de lâmpadas com potência menor do que o especificado é formado pelos pares (100,101) e (101,100), portanto a resposta é 2.

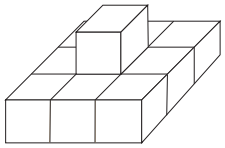
Pirâmide

Nome do arquivo: `piramide.c`, `piramide.cpp`, `piramide.pas`, `piramide.java`, `piramide.js` *ou* `piramide.py`

O rei da Nlogônia decidiu construir uma pirâmide no jardim do Palácio Real, usando cubos de pedra de mesmo tamanho. A *dimensão* de uma pirâmide é o número de cubos de pedra num dos lados da base (primeira camada) da pirâmide. A base da pirâmide é quadrada, ou seja, cada lado tem o mesmo número de cubos de pedra.

Na pirâmide, a partir da segunda camada, cada cubo de pedra deve ser empilhado exatamente em cima de outro cubo de pedra que não esteja na borda da camada abaixo. Além disso, o número de camadas deve ser o maior possível para uma dada dimensão, e em cada camada deve ser usado o maior número de cubos de pedra possível.

A figura abaixo à esquerda mostra uma pirâmide de dimensão 3; a figura à direita mostra o *plano de contrução* para essa pirâmide, indicando quantos cubos de pedra devem ser empilhados em cada posição.



1	1	1
1	2	1
1	1	1

O rei ainda não decidiu qual a dimensão da pirâmide que vai construir, mas como é muito detalhista já avisou os Arquitetos Reais que antes de iniciar a contrução eles devem produzir um plano de contrução para a dimensão escolhida.

Ajude os Arquitetos Reais, escrevendo um programa que, dada a dimensão da pirâmide, produza o seu plano de construção.

Entrada

A primeira e única linha da entrada contém um número inteiro N, a dimensão da pirâmide.

Saída

Seu programa deve produzir o plano de contrução da pirâmide, constituído por N linhas, cada linha contendo N números inteiros.

Restrições

- $1 \leq N \leq 100$

Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 10 pontos, $1 \leq N \leq 3$.

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**

`3`	`1 1 1`
	`1 2 1`
	`1 1 1`

Explicação do exemplo 1: Para uma pirâmide de dimensão 3, o maior número de camadas possível é 2.

Exemplo de entrada 2	**Exemplo de saída 2**
8	1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 1 2 3 3 3 3 2 1 1 2 3 4 4 3 2 1 1 2 3 4 4 3 2 1 1 2 3 3 3 3 2 1 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1

Explicação do exemplo 2: Para uma pirâmide de dimensão 8, o maior número de camadas possível é 4.