

X - Olimpíada de Programação de Computadores – (OPC – 2017)

LEIA ATENTAMENTE AS INSTRUÇÕES ABAIXO:

- 1) Utilize somente o material didático que sua equipe trouxe para a sala;
- 2) Todos os programas devem ser resolvidos usando C ou JAVA. Não poderá ser alternada a solução entre as linguagens, caso o primeiro problema, seja resolvido em C os demais não poderão ser resolvidos em JAVA e vice e versa.
- 3) Seus programas serão testados com outros valores, além dos indicados neste caderno de questões;
- 4) Utilize quando necessário: Estruturas de repetições, condições, scanf, getchar, printf ou putchar, vetores e matrizes.
- 5) Para seus programas serem avaliados copie-os para um arquivo no bloco de notas. Em seguida, grave-os num CD e entregue para o juiz da sala. **Cada arquivo** deve conter os nomes completos dos integrantes da sua equipe e o número da mesma, informado pelos juízes no início da olimpíada. Tais informações devem ser escritas no início de **cada programa em linhas de comentário (//.....);**
- 6) Não serão esclarecidas dúvidas, nem interpretados e/ou lidos os enunciados das questões.
- 7) **TODOS OS PROGRAMAS DESENVOLVIDOS** devem apresentar resultados na tela para o usuário, tais como as saídas demonstradas após o enunciado de cada exercício. Portanto, teste cada programa com os valores de entradas propostos.
- 8) Arquivos a serem enviados para os Juízes gravados em um CD.
- 10) Nome dos Arquivos fontes: probA.java, probA.c, probB.cpp
- 11) **Todas as saídas devem ser apresentadas depois de todas as entradas.**
- 12) **Não programe mensagens de entradas e saídas de dados, a menos que o enunciado do exercício exija.**
- 13) **Não poderam ser usados PEN-DRIVE, NOTEBOOK e CELULARES DURANTE A OLIMPÍADA.**

Problema – A (LEDs) (A.cpp,A.c,A.java)

O professor Alessandro, precisa montar em seu laboratório um painel contendo diferentes números de LEDs. Ele não tem certeza se o número de LEDs que possui será suficiente para completar o painel desejado. Considerando a configuração dos LEDs conforme a disposição abaixo. Faça um programa que ajude o professor Alessandro a descobrir o número de LEDs necessários para definir o valor dados.



Entrada

As entradas contêm um número inteiro **N**, ($1 \leq N \leq 2000$) correspondente ao número de casos de teste, seguido por **N** linhas, cada linha contendo um número de V, ($1 \leq V \leq 10^5$) correspondente ao valor que o professor Alessandro quer definir com os LEDs.

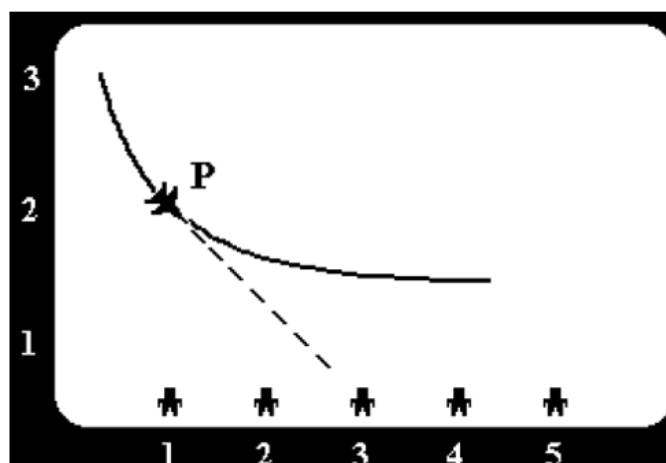
Saída

Para cada caso de teste, imprima uma linha contendo o número de LEDs que o professor Alessandro precisa para definir o valor desejado, seguido da palavra "Leds".

Exemplo de entrada	Saída para o exemplo de entrada
3	27 leds
115380	29 leds
2819311	25 leds
23456	

Problema – B (Jogo de Videogame) (B.cpp,B.c,B.java)

Em um jogo de videogame, os aviões voam da esquerda para direita, segundo a trajetória mostrada na figura, disparando suas balas na direção tangente, contra os alvos no eixo x. Escrever um programa que dados os valores calcular o trajeto tangente percorrida pelo tiro.



Entrada

A entrada consiste de um número inteiro N ($1 \leq N \leq 1000$) positivo que determinará a quantidade de entradas e a segunda e a terceira linha em diante dois números inteiros positivos A e B (1 a 10^6), onde (A) representa o caminho percorrido pelo avião e (B) a distância do vértice do tiro com a distância do vértice do avião.

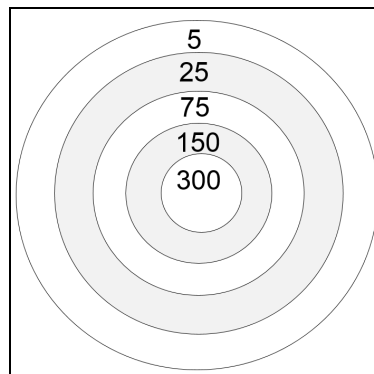
Saída

Saída deve ser composta de uma linha com um número real que representa o valor da linha tracejada na figura, que corresponde ao caminho percorrido pelo tiro.

Exemplo de entrada	Saída para o exemplo de entrada
3	3.46
4	8.49
2	4.00
9	
3	
5	
3	

Problema – C (O Alvo)
(C.cpp,C.c,C.java)

O atleta brasileiro Marcus Vinícius D’Almeida, de 20 anos, já foi considerado o segundo maior arqueiro do mundo, para realizar o seu treinamento, precisava confeccionar vários Alvos com diferentes pontuações, sabe-se que as pontuações encontradas em cada Alvo seguem a uma fatoração de números exemplo, para um Alvo de 300 pontos seus valores seriam 300,150,75,25 e 5, você deve escrever um algoritmo que ajude Marcus, ao determinar o número de pontos total, realizar a fatoração das demais pontuações.



Entrada

A entrada deverá ser uma única linha contendo o valor N máximo de pontos do Alvo onde N, $(1 \leq N \leq 10^3)$.

Saída

A saída deve ser composta por linhas, que expressão, os pontos fatorados nas áreas do Alvo, representados na figura acima.

Exemplo de entrada	Saída para o exemplo de entrada
6024	251
300	753
30	1506
	3012
	6024
	5
	25
	75
	150
	300
	5
	15
	30

Problema – D (Jogo de Xadrez) (D.cpp,D.c,D.java)

No jogo de xadrez, a torre é uma peça que pode se mover para qualquer outra posição do tabuleiro na linha ou na coluna da posição que ela ocupa. O professor Paulo está tentando inventar um novo tipo de jogo de xadrez onde todas as peças são torres, o tabuleiro também é quadrado mas pode ter qualquer dimensão e cada posição do tabuleiro é anotada com um número inteiro positivo, como na figura ao lado.

	1	2	3	4	5	6
1	4	1	3	8	4	5
2	9	2	8	9	2	7
3	5	5	4	3	2	5
4	8	2	9		9	8
5	7	1	3	2	1	2
6	5	1	2	9	3	8

Ele definiu o peso de uma posição $(i;j)$ como sendo a soma de todos os números que estejam na linha i com todos os números da coluna j , mas sem somar o número que está exatamente na posição $(i;j)$. Quer dizer, se uma torre estiver na posição $(i;j)$, o peso da posição é a soma de todas as posições que essa torre poderia atacar.

O professor Paulo está solicitando a sua ajuda para implementar um programa que determine qual é o peso máximo entre todas as posições do tabuleiro. No exemplo da figura acima, com um tabuleiro de dimensão seis (ou seja, seis linhas por seis colunas), o peso máximo é 67, referente à posição (4;4).

Entrada

A primeira linha da entrada contém um inteiro N , representando a dimensão do tabuleiro, na segunda e terceira linha a posição de referência da torre $(i;j)$. Cada uma das N linhas seguintes contém N inteiros positivos X_i , definindo os números em cada posição do tabuleiro.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o peso máximo entre todas as posições do tabuleiro.

Restrições

$$3 \leq N \leq 1000$$

$$0 < X_i \leq 100$$

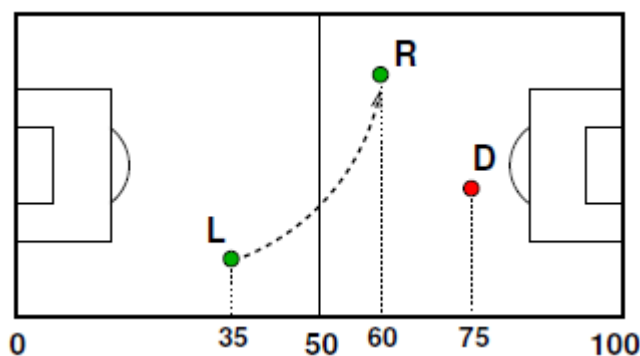
Exemplo de entrada	Saída para o exemplo de entrada
6 4 4 4 1 3 8 4 5 9 2 8 9 2 7 5 5 4 3 2 5 8 2 9 1 9 8 7 1 3 2 1 2 5 1 2 9 3 8	67

Exemplo de entrada	Saída para o exemplo de entrada
3	12
2	
2	
5 1 1	
5 2 1	
8 5 5	

Problema – E (Impedimento) (E.cpp,E.c,E.java)

A regra do impedimento no jogo de futebol é fundamental para identificação das posições de três jogadores: L o jogador atacante que lança a bola; R o jogador atacante que recebe a bola; e D o último jogador defensor. E a regra vale somente se o jogador R está no seu campo de ataque; se o jogador R está no seu campo de defesa ou na linha divisória do meio campo, ele não está em impedimento. Neste problema o campo tem 100 metros de comprimento. O técnico Tele Santana, precisava quantificar, o número de vezes que seus jogadores de ataque, encontravam-se em impedimento a cada ataque. Dadas as posições desses três jogadores, no momento exato do lançamento, haverá impedimento se e somente se a seguinte condição for verdadeira:

$$(R > 50) \text{ e } (L < R) \text{ e } (R > D)$$



A figura acima, mostra um exemplo onde não há impedimento.

Entrada

A primeira linha da entrada contém um inteiro N, representando o número de rodadas realizados em seguida a segunda linha com número de ataques na rodada, a terceira, quarta e quinta linhas, devem conter os valores de L, R e D sucessivamente.

Saída

Seu programa deve produzir linhas correspondentes ao número de rodadas, e se houve impedimento "S" e "N" caso não haja impedimento.

Restrições

$$\begin{aligned} 0 &\leq L \leq 100 \\ 0 &\leq R \leq 100 \\ 0 &\leq D \leq 100 \end{aligned}$$

Exemplo de entrada	Saída para o exemplo de entrada
2	1 N
3	1 S
35	1 N
60	
75	2 S
55	2 N
68	
67	1 N
80	1 S
80	1 N
80	
2	
55	
68	
66	1 S
66	
80	
81	
1	
3	
35	
60	
75	
55	
68	
67	
70	
80	
80	
1	
1	
55	
68	
67	