



Maratona POP

25 de outubro de 2018

(Este caderno contém 08 problemas)

A PROVA TERÁ DURAÇÃO DE **TRÊS HORAS**

LEIA ATENTAMENTE AS INSTRUÇÕES ABAIXO ANTES DE INICIAR A PROVA

- A prova deve ser realizada em equipe de até 03 pessoas;
- Observem o nome do arquivo que deve ser enviado para cada problema;
- Cada questão tem um tempo limite para execução;
- É permitido consultar material impresso durante a prova;
- Não é permitida a consulta de qualquer material online;
- Este caderno de tarefas é composto de 10 páginas;
- Verifique se o caderno está completo.

Problema A - Melhor Time

Nome do Programa: melhortime.(c|cpp|pas|py|java)

Tempo: 2 segundos

Atenção! Atenção! O servidor que julga o time vencedor da maratona POP enlouqueceu! Ele não consegue mais computar qual time foi o vencedor da competição! Já sei! Já sei! O time que conseguir resolver o problema do servidor vai ganhar um balão! =-)

Entrada

A primeira linha da entrada contém um número inteiro N ($1 \leq N \leq 200$) que representa o número de times inscritos na competição. As próximas N linhas apresentam o seguinte formato: $x_n y_n$, sendo x_n um número inteiro que identifica o time e y_n um número inteiro que informa a pontuação obtida por este time. ($0 \leq x_n, y_n \leq 200$).

Saída

A saída deve ser o número identificador do time que obteve mais pontos. Obs.: Todos os times obtiveram pontuações diferentes.

Casos de Teste

Entrada	Saída
5 0 1 1 4 2 3 3 2 4 0	1

Entrada	Saída
4 0 1 1 2 2 3 3 4	3

Problema B - SpyRobot #1

Nome do Programa: spyrobot.(c|cpp|pas|py|java)

Tempo: 2 segundos

O famoso espião brasileiro 00171 recebeu a missão de descobrir a senha de uma renomada senhora da alta sociedade pessoense. Para cumprir sua tarefa, 00171 recebeu um robô espião. Este robô é capaz de dizer qual tecla foi apertada apenas ouvindo os ruídos que a tecla faz ao ser pressionada.

00171, porém, estava receoso e resolveu testar a eficiência do robô antes de utilizá-lo. Seu trabalho é escrever um programa que calcule o número de acertos que o robô teve no ambiente de treinos montado por 00171.

Entrada

A entrada contém vários casos de teste. A primeira linha da entrada tem um inteiro T ($1 \leq T \leq 100$) indicando o número de testes. Cada teste possui 3 linhas. A primeira linha tem um inteiro N ($1 \leq N \leq 60$) que representa o número de teclas apertadas. A segunda linha tem N números (de 0 a 9) que representam as teclas apertadas por 00171. A terceira linha possui N números (de 0 a 9) representando os palpites do robô sobre as teclas pressionadas por 00171.

Saída

A saída deve ser o número de acertos do robô.

Casos de Teste

Entrada	Saída
2	4
4	2
0 1 2 3	
0 1 2 3	
10	
0 0 9 9 8 7 8 7 1 1	
0 4 6 7 1 0 8 6 5 6	

Problema C - Ordenado

Nome do Programa: ordenado.(c|cpp|pas|py|java)

Tempo: 2 segundos

Timério é um professor de matemática muito metódico. Todos os dias ele acorda exatamente às 5:30, escova os dentes, prepara seu café da manhã e sai caminhando em direção ao IFPB. Chegando à escola (exatamente 1 minuto antes do início de sua aula), Timério pede que os alunos formem uma fila antes de entrar na sala. Como é muito metódico, pede que um aluno maior nunca fique à frente de um menor na fila. Como a vista do antiquíssimo professor Timério não está mais lá essas coisas, ele pede que você escreva um programa para dizer se a fila está de acordo com o método do professor.

Dada a altura dos alunos na fila, do primeiro ao último, seu programa deve dizer se a fila obedece ou não os critérios do professor Timério.

Entrada

A entrada terá exatamente duas linhas no seguinte formato:

N

$x_1 \ x_2 \ x_3 \ x_4 \ \dots \ x_n$

O número inteiro $1 \leq N \leq 200$ indica o número de alunos na fila. Em seguida, serão informadas as alturas dos N alunos, na ordem em que se encontram na fila. A altura de cada aluno é representada por um número inteiro $1 \leq x \leq 100$.

Saída

Caso a fila obedeça aos critérios do professor Timério, o programa deve imprimir SIM. Caso contrário, deve imprimir NAO.

Casos de Teste

Entrada	Saída
10 1 2 3 4 5 6 7 8 9 10	SIM

Entrada	Saída
8 1 1 2 2 3 3 4 4	SIM

Entrada	Saída
12 1 1 2 2 2 3 3 3 2 4 5 6	NAO

Problema D - Aritmética com palavras

Nome do Programa: aritmetica.(c|cpp|pas|py|java)

Tempo: 2 segundos

Um arqueólogo encontrou vestígios de uma antiga civilização que não descobriu os números, mas descobriu a matemática! Eles usavam a escrita dos números e dos sinais para fazer as operações. Como esperado, contudo, eles só conseguiam realizar operações simples com números de um a vinte. Sua tarefa é interpretar expressões desta civilização e imprimir o resultado.

Entrada

A entrada corresponde a uma expressão. Cada expressão possui dois ou mais números e uma ou mais operações. As operações eram a soma, representada pela palavra “mais”, a subtração, representada pela palavra “menos”, e a multiplicação, representada pela palavra “vezes”. Lembre que a multiplicação sempre é realizada antes das adições e subtrações. Obs.: o número “14” era escrito tanto como “catorze” quanto como “quatorze”, mas deverá ser escrito como “catorze” na saída. A resposta nunca será menor que um nem maior que vinte, assim como os números da entrada.

Saída

A saída deve ser o resultado da operação.

Casos de Teste

Entrada	Saída
cinco mais sete	doze

Entrada	Saída
sete mais nove menos dois	catorze

Entrada	Saída
vinte menos dois vezes nove	dois

Entrada	Saída
um mais um mais um vezes dois menos um	três

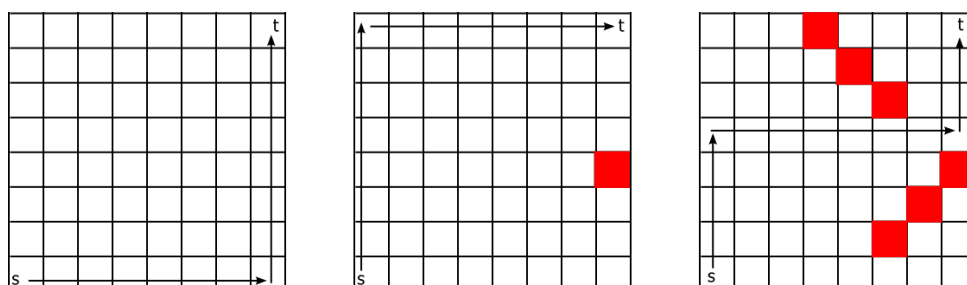
Problema E - Sem Curvas

Nome do Programa: curvas.(c|cpp|pas|py|java)

Tempo: 2 segundos

Nelson é uma pessoa comum, um cidadão comum e um trabalhador comum. Nelson acorda todos os dias às 7 horas, faz o que tem que fazer, pega seu carro e dirige até o trabalho. Nelson está muito feliz na sua rotina. A única coisa que Nelson odeia é fazer curvas!

A cidade de Nelson foi planejada de modo que qualquer motorista só pode ir reto, virar 90 graus para esquerda ou 90 graus para a direita. Desta forma, a cidade de Nelson pode ser representada como uma grade. Nesta grade, a casa de Nelson fica no canto inferior esquerdo (s) e seu trabalho fica no canto superior direito (t).



Em dias normais, Nelson precisa fazer uma única curva para ir de casa para o trabalho. Mas é muito comum que pontos de sua cidade estejam interditados por estarem em obras. Nelson pediu sua ajuda para calcular a rota que faça menos curvas uma vez que ele sabe todos os pontos da cidade que estão interditados.

Entrada

A entrada representa a cidade de Nelson. A primeira linha tem um número inteiro N ($1 \leq N \leq 100$), representando o número de linhas e de colunas da cidade. As próximas N linhas contêm N inteiros cada. Um número 0 na posição (i, j) , indica que esta rota está livre, enquanto um número 1 na posição (i, j) indica que esta está interditada.

Saída

A saída do programa deve ser a menor quantidade de curvas que Nelson deve fazer para chegar ao seu trabalho. Nelson pode sair de casa horizontalmente ou verticalmente e isso não conta como uma curva.

Casos de Teste

Entrada	Saída
8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	1

Entrada	Saída
8 00000000 00000000 00000000 00000000 00000001 00000000 00000000 00000000 00000000	1

Entrada	Saída
8 00010000 00001000 00000100 00000000 00000001 00000010 00000100 00000000	2

Problema F - Um pescador com T.O.C.

Nome do Programa: toc.(c|cpp|pas|py|java)

Tempo: 2 segundos

Zé pinto foi um jogador de futebol à frente do seu tempo, atacava e defendia como poucos! Foi campeão belo-horizontino, mineiro, sudestino, brasileiro, sul-americano, pan-americano e mundial. Mas a idade chega para todos, não é? Depois da aposentadoria, o famoso camisa 7 do Sete de Setembro virou pescador. O seu único problema é seu T.O.C. (transtorno obsessivo compulsivo) com o número 7. Ele só consegue voltar para casa com exatamente 7 quilos de peixe. Caso não consiga, devolve todos para a água e volta para casa de mãos abanando. Dado o peso de todos os peixes do lago onde Zé Pinto foi pescar, sua tarefa é dizer se é possível que ele volte para casa com exatamente 7 quilos de peixe.

Entrada

A entrada terá apenas uma linha no seguinte formato:

$N \ x_1 \ x_2 \ x_3 \ x_4 \ \dots \ x_n$

O número inteiro N ($1 \leq N \leq 50$) indica o número de peixes no lago. Em seguida, serão informados os pesos dos N peixes. O peso de cada peixe é representado por um número inteiro $1 \leq x \leq 7$.

Saída

Caso exista um conjunto de peixes cuja soma de pesos seja 7, o programa deve imprimir SIM. Caso contrário, deve imprimir NAO.

Casos de Teste

Entrada	Saída
10 1 2 3 4 5 6 7 1 2 3	SIM

Entrada	Saída
9 1 1 1 1 1 1 1 1 1	SIM

Entrada	Saída
15 2 3 3 3 3 3 3 3 3 3 3 3 3 3	NAO

Problema G - Flash Mob

Nome do Programa: flashmob.(c|cpp|pas|py|java)

Tempo: 2 segundos

Segundo a Wikipedia, *Flash Mobs* são aglomerações instantâneas de pessoas em certo lugar para realizar determinada ação inusitada previamente combinada, estas se dispersando tão rapidamente quanto se reuniram. O grande Phinerilson, o Cupido, codinome Ps2C, é um famoso organizador de *flash mobs* com solteiros com o objetivo de formar casais. Funciona de maneira bem simples, Ps2C convida N mulheres e N homens para a festa, exibe as fotos dos homens e pede para que elas marquem aqueles com quem formariam um casal. O dever de Ps2C é apenas organizar os casais para que ninguém fique solteiro.

Dessa vez, no entanto, Ps2C convidou pessoas demais e não está conseguindo verificar se é ou não é possível que ninguém fique solteiro. Sua missão é ajudar Phinerilson a resolver este problema.

Entrada

A entrada contém apenas um caso de teste, mas seu programa será testado com várias entradas diferentes. A primeira linha da entrada tem um inteiro N ($1 \leq N \leq 5000$) indicando o número de mulheres (e de homens) que foram convidados. As N linhas seguintes trazem listas de inteiros (separadas por espaços) que indicam os *IDs* dos homens que as mulheres marcaram como possíveis pares. Por exemplo, se a linha k tem os inteiros 4 8 9, isso significa que a senhorita com *ID* k marcou os rapazes com *IDs* 4, 8 e 9 como possível casal. Os *IDs* dos homens são números inteiros de 0 a $N-1$, assim como o das mulheres.

Saída

Caso seja possível formar casais de modo que ninguém fique solteiro, a saída deve ser s2. Caso contrário, s/2.

Casos de Teste

Entrada	Saída
4 0 1 2 3	s2
Entrada	Saída
4 0 1 2 3 1 1	s/2

Problema H - Árvore de busca binária ótima¹

Nome do Programa: binaria.(c|cpp|pas|py|java)

Tempo: 2 segundos

É dado um conjunto $S = (e_1, e_2, \dots, e_n)$ de n elementos distintos tal que $e_1 < e_2 < \dots < e_n$. Considerando uma árvore de busca binária dos elementos de S , é desejado que quanto maior a frequência em que um elemento é buscado, mais perto este esteja da raiz.

Deste modo, o custo de acessar um elemento e_i de S em uma árvore ($cost(e_i)$) é o número de arestas no caminho que conecta o nó raiz a este elemento.

Dadas as frequências de busca dos elementos de S , ($f(e_1), f(e_2), \dots, f(e_n)$), dizemos que o custo total de uma árvore é:

$$f(e_1) * cost(e_1) + f(e_2) * cost(e_2) + \dots + f(e_n) * cost(e_n)$$

Deste modo, a árvore com o menor custo total é aquela com a melhor representação para busca dos elementos de S . Por conta disto, ela é chamada de árvore de busca binária ótima.

Entrada

A entrada contém várias instâncias, uma por linha. Cada linha começa com um número $1 \leq n \leq 250$, indicando o tamanho de S . Seguem n números, na mesma linha, representando as frequências de busca de cada elemento de S : $f(e_1), f(e_2), \dots, f(e_n)$, $0 \leq f(e_i) \leq 100$. A entrada termina com EOF.

Saída

Para cada instância da entrada, você deve imprimir uma linha na saída contendo o custo total da árvore de busca binária ótima.

Casos de Teste

Entrada	Saída
1 5	0
3 10 10 10	20
3 5 10 20	20
4 5 10 20 20	40

¹ Baseado no problema 10304 do site uva.onlinejudge.org.