

lista de vetores - 2017.04.24

Prof. Dr. Gustavo Teodoro Laureano
Profa. Dra. Luciana Berretta
Prof. Dr. Thierson Rosa Couto

Sumário

1	Acumulado de Elementos	2
2	Counting Sort	4
3	Distância entre pontos	5
4	Elementos Únicos	6
5	Frequência e Maior	7
6	Intercala	8
7	Ordena Lista	9
8	Loteria	10
9	Maior Frequencia	11
10	Máxima Coordenada	12
11	Mediana	13
12	Menor Distancia	14
13	Pares e Ímpares	16
14	Apague e Ganhe	17
15	CPF	18
16	Conversão de Decimal para Binário	19
17	Contagem de Elementos Únicos	20
18	Imprimir Um Vetor na Ordem Inversa	21
19	Ler e Imprimir	22
20	Os Verdadeiros Sete Anões da Branca de Neve	23

1 Acumulado de Elementos



(++)

Faça um programa que receba vários vetores de números inteiros, calcule o maior elemento (M) de cada vetor e apresente a frequência dos valores menores ou iguais a i , onde $i = 0, 1, 2, \dots, M$.

Entrada

O programa possui vários casos de testes. A primeira linha de cada caso contém um inteiro $1 < N \leq 10000$, representando o tamanho do vetor. A segunda linha conterá N inteiros entre 0 e 10000, representando os N elementos do vetor. A entrada termina quando $N = 0$.

Saída

O programa gera várias linhas de saída para cada entrada. Cada linha apresenta o valor entre parênteses seguido de um espaço em branco e a quantidade de números que são menores ou iguais a esse valor, seguido de '\n'.

Exemplo

Entrada										
10										
6	13	7	3	13	6	14	3	14	9	
5										
9	8	7	6	5						
8										
0	1	2	3	4	5	6	7			
0										
Saída										
(0)	0									
(1)	0									
(2)	0									
(3)	2									
(4)	2									
(5)	2									
(6)	4									
(7)	5									
(8)	5									
(9)	6									
(10)	6									
(11)	6									
(12)	6									
(13)	8									
(14)	10									
(0)	0									
(1)	0									
(2)	0									
(3)	0									
(4)	0									
(5)	1									
(6)	2									
(7)	3									
(8)	4									
(9)	5									
(0)	1									
(1)	2									
(2)	3									
(3)	4									
(4)	5									
(5)	6									
(6)	7									
(7)	8									

2 Counting Sort



(++++)

Counting sort é um algoritmo de ordenação estável cuja complexidade é $O(n)$. As chaves podem tomar valores entre 0 e $M-1$. Se existirem k_0 chaves com valor 0, então ocupam as primeiras k_0 posições do vetor final: de 0 a k_0-1 .

O procedimento para implementação do Counting Sort segue os seguintes passos:

1. Cria-se um vetor $vCount[M+1]$ e $vOrd[N-1]$, onde N é a quantidade de elementos a serem ordenados e M é o maior valor entre os elementos a serem ordenados.
2. Inicializa-se todas as posições de $vCount$ com 0.
3. Percorre-se o vetor v e, para cada posição i de v faz-se $vCount[v[i]]++$, o que faz com que, no final, cada posição i de $vCount$ contem a quantidade de vezes que a chave i aparece em V .
4. Acumula-se em cada elemento de $vCount$ o elemento somado ao elemento anterior, desta forma, $vCount[i]$ indica a posição-1 ordenada do primeiro elemento de chave i .
5. Guarda-se em $vOrd$ os valores de V ordenados de acordo com $vOrd[vCount[v[i]-1]=V[i]$. E decrementa-se $vCount[v[i]]$ de uma unidade.
6. Copia-se $vOrd$ para v .

Esta implementação tem a desvantagem de precisar de vetores auxiliares. O Counting Sort ordena exclusivamente números inteiros pelo fato de seus valores servirem como índices no vetor de contagem.

Entrada

O programa possui vários casos de testes. A primeira de cada caso contém um inteiro N , $1 < N \leq 10000$, representando o tamanho do vetor. A segunda linha conterá N inteiros entre 0 e 1000, representando os N elementos do vetor. A entrada termina quando $N=0$.

Saída

O programa gera uma linha de saída para cada entrada, contendo os valores recebidos na entrada ordenados de acordo com o Counting Sort. Entre cada valor há um espaço em branco. Antes do primeiro valor não deve-se imprimir nada e após ao último valor deve-se apenas quebrar uma linha.

Exemplo

Entrada
10
6 13 7 3 13 6 14 3 14 9
5
9 8 7 6 5
8
0 1 2 3 4 5 6 7
0
Saída
3 3 6 6 7 9 13 13 14 14
5 6 7 8 9
0 1 2 3 4 5 6 7

3 Distância entre pontos



(++)

Faça um programa que leia vários pares de pontos e calcule a distância entre eles. Considere que a distância entre dois pontos $A(a_x, a_y, a_z)$ e $B(b_x, b_y, b_z)$ é calculada como:

$$d(A, B) = |BA| = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2} \quad (1)$$

Entrada

A entrada consiste de várias linhas. A primeira linha apresenta um número de pontos N , com $2 \leq N \leq 1000$. As N linhas seguintes apresentam pontos no espaço na forma x y z , com x , y e z números reais tais que $-1000 \leq x, y, z \leq 1000$. Faça um programa que calcule a distância entre dois pontos consecutivos nesta lista. Note que, com exceção do primeiro e último valor de entrada, todos os pontos serão utilizados duas vezes, uma para o cálculo de distância com o ponto que veio antes na lista e outra para o ponto que veio depois.

Saída

A saída consiste de $(N-1)$ linhas, cada uma contendo a distância entre os pontos com 2 casas decimais após a vírgula. Após a impressão do último valor, quebre uma linha.

Exemplo

Entrada
2
4 1 0
-1 2 1
Saída
5.20

Entrada
4
1 1 5
2 -1 3
4 2 -1
-3 4 2
Saída
3.00
5.39
7.87

Entrada
4
15.89 0.7 0.53
0.45 0.38 0.22
0 0 0
0 0 1
Saída
15.45
0.63
1.00

4 Elementos Únicos



(+++)

Dada uma sequência de N números inteiros na ordem crescente, identifique os elementos únicos.

Entrada

Na primeira linha há um inteiro N, $1 \leq N \leq 1000$, representando a quantidade de elementos. Nas N linhas seguintes haverá um número inteiro (positivo ou negativo) por linha.

Saída

O programa apresenta um sequência de elementos únicos, na ordem crescente. Cada elemento é apresentado em uma linha.

Exemplo

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

Entrada
5
9
9
9
9
9
Saída
9

Entrada
7
4
4
4
4
4
4
8
Saída
4
8

Entrada
1
900
Saída
900

Entrada
3
-200
-200
-10
Saída
-200
-10

5 Frequência e Maior



(++)

Dada uma sequência de N notas entre 0 e 10, escreva um programa que exiba o valor da última nota informada e quantas vezes ela apareceu no conjunto. O programa deve exibir ainda a maior nota informada e a posição (índice do vetor) da sua primeira ocorrência.

Entrada

Na primeira linha há um inteiro N, sendo $1 \leq N \leq 10000$ representando a quantidade de notas da sequência. Não é necessário validar o valor de N na entrada. Nas N linhas seguintes haverá um número inteiro entre 0 e 10, inclusive, em cada linha.

Saída

O programa gera 2 linhas de saída. A primeira linha exibirá a frequência da última nota informada e a segunda linha exibirá a maior nota e a posição (índice do vetor) da sua primeira ocorrência, seguindo o formato da saída apresentado a seguir. Não se esqueça de quebrar uma linha após a última impressão.

Exemplo

Entrada
11
5
6
3
4
3
8
7
4
8
6
4
Saída
Nota 4, 3 vezes
Nota 8, indice 5

6 Intercala



(+++++)

Faça um algoritmo que aloque dois vetores V1 e V2 com o tamanho de cada entrada q1 e q2, receba os q1 valores no vetor V1 e os q2 valores no vetor V e construa um terceiro vetor, Vr, com a intercalação dos vetores V1 e V2 de forma ordenada.

Entrada

A entrada consiste de dois número positivo q1 e q2 , sendo $0 < q(1,2) \leq 500000$, representando a quantidade de entradas do programa. Seguido de q1 +q2 linhas, onde nas q1 primeiras linhas estão os q1 valores e nas demais q2 linhas estão os q2 valores. Esses valores são naturais n, $0 \leq n \leq 999999$. E ainda, dentro do mesmo bloco é garantido que o número n representado na linha q é menor que o número que está em q+1 e maior que ou igual ao que está em q-1. Ou seja: $n(q-1) \leq n(q) < n(q+1)$ para todo q.

Saída

A saída deverá ser todos os q1 +q2 valores das duas entradas intercalados e impressos de forma crescente.

Exemplo

Entrada
5
7
1
3
5
7
21
0
2
4
6
8
10
12
Saída
0
1
2
3
4
5
6
7
8
10
12
21

7 Ordena Lista



(++)

Faça um programa para imprimir uma lista de inteiros em ordem crescente.

Entrada

A entrada contém apenas um caso de teste. Na primeira linha há um inteiro N , $1 < N \leq 1000$, representando a quantidade de inteiros que serão informados. Em seguida haverá N linhas com um inteiro em cada linha.

Saída

Seu programa gera N linhas de saída, contendo em cada linha um inteiro que são os mesmos informados na entrada, porém em ordem crescente. Após o último número impresso, quebre uma linha.

Exemplo

Entrada
5
7
1
3
4
5
Saída
1
3
4
5
7
Entrada
2
4
3
Saída
3
4

8 Loteria



(++++)

A Mega-Sena é a maior loteria do Brasil. Para ganhar o prêmio máximo é necessário acertar a sena, o que significa obter coincidência entre seis dos números apostados e os seis números sorteados, de um total de sessenta dezenas (de 01 a 60), independentemente da ordem da aposta ou da ordem do sorteio. O concurso prevê também a chance de ganhar parte do prêmio, acertando a quina ou a quadra. A Mega-Sena foi lançada em março de 1996 e já premiou mais de 200 ganhadores na faixa principal. Os prêmios correspondem a 32,2% da renda das apostas ao imposto de renda correspondem 13,8% de todas as apostas. Os vencedores têm 90 dias para retirar o prêmio, se o período expirar, o dinheiro do prêmio será transferido ao Tesouro Nacional e investido em programas educacionais. Vale lembrar que a probabilidade de acerto em uma única aposta de 6 dezenas é de 1 em 50.063.860, o que representa um percentual de 0,000002%. Faça um programa que receba todas as apostas e as seis dezenas sorteadas de um concurso e mostre quantos vencedores para sena, quina e quadra houve.

Entrada

Na primeira linha da entrada haverá uma linha com as seis dezenas sorteadas, separadas por um espaço em branco cada. Na linha seguinte haverá um inteiro N , $1 \leq N \leq 50000$, representando a quantidade de apostas. Em seguida, em cada uma das N linhas haverá as seis dezenas de cada aposta, sendo que as dezenas estão no intervalo entre 1 e 60 e sem repetição de dezenas por apostas.

Saída

A saída consiste de 3 linhas contando uma das seguintes frases: “Houve K acertador(es) da sena” ou “Houve K acertador(es) da quina” ou ainda “Houve K acertador(es) da quadra”, onde K é quantidade de acertadores para a faixa. Caso não haja acertadores a seguinte frase deve ser apresentada: “Nao houve acertador para sena” ou “Nao houve acertador para quina” ou ainda “Nao houve acertador para quadra”. Ao exibir a última frase quebre uma linha.

Exemplo

Entrada
23 12 33 19 10 8
5
23 19 8 12 60 18
14 60 12 44 54 10
8 3 12 19 33 10
33 15 7 60 12 10
22 12 19 23 33 11
Saída
Nao houve acertador para sena
Houve 1 acertador(es) da quina
Houve 2 acertador(es) da quadra

9 Maior Frequencia



(++)

Dada uma sequência de N números entre 0 e 100. Determine qual o valor de maior frequência. Caso haja mais de um valor tenha a maior frequência, mostre o menor deles.

Entrada

Na primeira linha há um inteiro N , $1 \leq N \leq 1000000$, representando a quantidade números. Nas N linhas seguintes haverá um número natural entre 0 e 100 inclusive por linha.

Saída

O programa gera apenas duas linhas. Na primeira dela mostre qual foi o valor com maior frequência. E na segunda linha, mostre a quantidade de vezes que esse número apareceu na sequência de valores. Após a impressão deste último valor quebre uma linha. Caso haja mais de um valor tenha a maior frequência, mostre o menor deles.

Exemplo

Entrada
10
1
7
4
29
7
4
7
8
7
29
Saída
7
4

10 Máxima Coordenada



(+++)

Faça um programa que leia vários pares de pontos, calcule o vetor definido entre eles e imprima a coordenada do vetor que possui o maior valor absoluto (módulo). Considere que o vetor que liga dois pontos A (x_1, y_1, z_1) e B (x_2, y_2, z_2) é calculada como: $BA = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$

Entrada

A entrada consiste de várias linhas. A primeira linha apresenta um número de pontos N, com $2 \leq N \leq 1000$. As N linhas seguintes apresentam pontos no espaço na forma x y z, com x, y e z números reais tais que $-1000 \leq x, y, z \leq 1000$. Faça um programa que calcule o vetor que liga dois pontos consecutivos nesta lista e imprima a coordenada de maior valor absoluto. Note que, com exceção do primeiro e último valor de entrada, todos os pontos serão utilizados duas vezes, uma para o cálculo do vetor com o ponto que veio antes na lista e outra para o ponto que veio depois.

Saída

A saída consiste de (N-1) linhas, cada uma contendo o módulo do valor da coordenada de maior valor absoluto, com 2 casas decimais após a vírgula. Após a impressão do último valor, quebre uma linha.

Exemplo

Entrada
2
4 1 0
-1 2 1
Saída
5.00

Entrada
4
1 1 5
2 -1 3
4 2 -1
-3 4 2
Saída
2.00
4.00
7.00

Entrada
4
15.89 0.7 0.53
0.45 0.38 0.22
0 0 0
0 0 1
Saída
15.44
0.45
1.00

11 Mediana



(++)

Em teoria da probabilidade e estatística, a mediana, é uma medida de localização do centro da distribuição dos dados, definida do seguinte modo: Ordenados os elementos da amostra, a mediana é o valor (pertencente ou não à amostra) que a divide ao meio, isto é, 50% dos elementos da amostra são menores ou iguais à mediana e os outros 50% são maiores ou iguais à mediana. Para uma coleção de tamanho ímpar, a mediana é exatamente o elemento médio, ou seja, aquele que a divide de acordo com a regra citada. Já para uma coleção de tamanho par, a mediana é determinada como a média aritmética dos dois elementos centrais.

Entrada

A entrada consiste de um único caso de teste. Na primeira linha, é informado um inteiro N , $0 < N \leq 10^6$, representando a quantidade de elementos da amostra de dados. Nas N linhas seguintes é informado um inteiro por linha, este valor varia de -2^{32} a $2^{32} - 1$.

Saída

A saída consiste da mediana dos dados informados. O valor da mediana deve ser formatado com duas casas decimais.

Exemplo

Entrada
6
1
3
4
5
4
2
Saída
3.50
Entrada
7
3
9
1
5
4
7
1
Saída
4.00

12 Menor Distancia



(++)

Calcular a distância entre os dois elementos mais próximos em uma sequência de N números inteiros.

Entrada

Na primeira linha há um inteiro T , $1 \leq T \leq 10$, representando a quantidade de testes a serem realizados. Para cada teste são esperados: (a) um número inteiro N , $2 \leq N \leq 1000$, que é o tamanho do vetor, e (b) uma sequência de N números inteiros que são os elementos do vetor. No vetor cada elemento deve estar no intervalo $[-1000, 1000]$.

Saída

O programa apresenta, para cada teste, o distância entre os dois elementos mais próximos e o número de comparações realizadas para resolver o problema. Observe que o número de comparações requeridas para resolver o problema para um vetor de tamanho N é N^2 .

Exemplo

Entrada
4
6
1 2 3 4 5 6
10
1 1 1 1 1 1 1 1 1 1
4
20 40 21 100
6
10 20 30 40 50 55
Saída
1 36
0 100
1 16
5 36

Entrada
3
20
0 0 0 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
20
90 80 70 60 50 40 30 20 10 0 1 100 200 300 400 500 600 700 800 900
20
90 80 70 60 50 40 30 20 10 0 100 200 300 400 500 600 700 800 900 1000
Saída
0 400
1 400
10 400

Entrada
1
2
0 1000
Saída
1000 4

Entrada
1
2
-1000 1000
Saída
2000 4

13 Pares e Ímpares



(+++)

Considerando a entrada de valores inteiros não negativos, ordene estes valores segundo o seguinte critério: • Primeiro os Pares • Depois os Ímpares Sendo que deverão ser apresentados os pares em ordem crescente e depois os ímpares em ordem decrescente.

Entrada

A primeira linha de entrada contém um único inteiro positivo N ($1 < N < 10^5$) que corresponde ao número de linhas de entrada que vêm logo a seguir. As próximas N linhas conterão, cada uma delas, um valor inteiro não negativo.

Saída

Apresente todos os valores lidos na entrada segundo a regra descrita acima. Cada número deve ser impresso em uma linha. Após imprimir último número quebre a linha.

Exemplo

Entrada
10
4
32
34
543
3456
654
567
87
6789
98
Saída
4
32
34
98
654
3456
6789
567
543
87

14 Apague e Ganhe



(++++)

Juliano é fã do programa de auditório Apagando e Ganhando, um programa no qual os participantes são selecionados através de um sorteio e recebem prêmios em dinheiro por participarem. No programa, o apresentador escreve um número de n dígitos em uma lousa. O participante então deve apagar uma certa quantidade de dígitos do número que está na lousa; o número formado por exatamente d dígitos que restaram é então o prêmio do participante. Juliano finalmente foi selecionado para participar do programa, e pediu que você escrevesse um programa que, dados o número que o apresentador escreveu na lousa, e quantos dígitos d devem restar na lousa, determina o valor do maior prêmio que Juliano pode ganhar.

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém dois inteiros n e d ($1 \leq d < n \leq 10^5$), indicando a quantidade de dígitos do número que o apresentador escreveu na lousa e quantos dígitos devem restar do número, após Juliano apagar alguns dígitos do número dado. A linha seguinte contém o número escrito pelo apresentador, que não contém zeros à esquerda. O final da entrada é indicado por uma linha que contém apenas dois zeros, separados por um espaço em branco.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha na saída, contendo o maior prêmio que Juliano pode ganhar.

Exemplo

Entrada
4 2
3759
6 3
123123
7 4
1000000
0 0
Saída
79
323
1000

15 CPF



(+++)

Você foi contratado pelas Indústrias Udilandenses (INUDIL) para desenvolver uma maneira de verificar se o Cadastro de Pessoa Física (CPF) indicado por um cliente era válido ou não. Conversando com amigos, você chegou à conclusão de que um CPF seria válido se a soma de todos os seus dígitos resultasse em número múltiplo de 11. Após verificação minuciosa, você descobriu que essa maneira só funciona em cerca de 80% dos casos, e você precisa de mais do que isso para garantir a qualidade do seu trabalho. Após pesquisar mais, você descobriu que dos 11 dígitos do CPF, os dois últimos são verificadores e dependem dos 9 dígitos anteriores. Vamos introduzir alguma notação. Considere um CPF com os seguintes dígitos

$$a_1a_2a_3a_4a_5a_6a_7a_8a_9 - b_1b_2$$

Para descobrirmos o dígito b_1 , procedemos da seguinte maneira: multiplicamos o primeiro dígito por 1, o segundo por 2, o terceiro por 3, o quarto por 4 e vamos assim até multiplicarmos o nono por 9. Então, somamos tudo isto. Após termos somado tudo, dividimos por 11. O dígito b_1 será o resto da divisão (ou 0, caso o resto seja 10).

Para o segundo dígito verificador, temos o seguinte: multiplicamos o primeiro elemento por 9, o segundo por 8, o terceiro por 7, o quarto por 6 e vamos assim até multiplicarmos o nono por 1. Então, somamos tudo isto e dividimos por 11. O dígito b_2 será o resto da divisão (ou 0, caso o resto seja 10).

Sabendo que isso vale para 100% dos CPFs, sua missão é implementar um programa que, dado um CPF, diga se ele é válido ou não.

Entrada

A entrada conterà uma linha com um inteiro T , que indica o número de casos de testes. Esta linha é seguida por T linhas, cada uma contendo uma sequência de 11 dígitos decimais, separados entre si por um espaço. Após o último dígito decimal segue o caractere de quebra de linha.

Saída

Para cada candidato a CPF da entrada, escreva "CPF valido", se ele for um CPF válido e, escreva "CPF invalido", em caso contrário.

Exemplos

Entrada
5
0 4 8 8 5 6 8 2 9 6 3
7 3 3 1 8 4 6 8 0 9 6
2 2 7 5 1 8 4 7 1 0 8
0 9 2 8 4 4 8 4 2 8 6
0 9 8 4 4 7 8 9 5 5 5
Saída
CPF invalido
CPF valido
CPF invalido
CPF valido
CPF invalido

16 Conversão de Decimal para Binário



(++)

Escreva um programa que leia vários números inteiros positivos na base decimal e escreva os valores correspondentes desses números na base binária.

Entrada

A entrada contém várias linhas, cada uma contendo um valor inteiro N na base decimal tal que seu número binário equivalente possui no máximo 32 bits. A saída termina por fim de arquivo.

Saída

A saída é formada por tantas linhas quantas forem as linhas na entrada. Cada linha contém a conversão para a base binária do valor em decimal que aparece na linha correspondente da entrada.

Exemplo

Entrada
0
1
2
3
5
572
Saída
0
1
10
11
101
1000111100

17 Contagem de Elementos Únicos



(+)

Elabore um programa que conte o número total de elementos únicos em um vetor de números inteiros.

Entrada

A entrada contém duas linhas. A primeira, contém um valor inteiro $n < 5000$ que corresponde ao número de elementos que aparecem na segunda linha. A segunda linha contém n valores inteiros, separados entre si por um espaço.

Saída

A saída é formada por uma linha contendo um valor inteiro que corresponde ao número de elementos que aparecem apenas uma vez no vetor. Após o valor, o programa deve imprimir o caractere de quebra de linha.

Exemplo

Entrada
7
3 6 2 9 2 7 9
Saída
3

18 Imprimir Um Vetor na Ordem Inversa



(+)

Escreva um programa em C para armazenar n valores inteiros em um vetor, e depois imprimi-los na ordem inversa a qual foram lidos.

Entrada

A entrada contém duas linhas. A primeira, contém um valor inteiro $n < 5000$ que corresponde ao número de elementos que aparecem na segunda linha. A segunda linha contém n valores inteiros, separados entre si por um espaço.

Saída

A saída é formada por uma linha contendo os n na ordem inversa da qual foram lidos.

Exemplo

Entrada
7
3 6 2 9 2 7 9
Saída
9 7 2 9 2 6 3

19 Ler e Imprimir



(+)

Escreva um programa em C para ler n elementos inteiros em um vetor, e depois imprimi-los.

Entrada

A entrada contém duas linhas. A primeira, contém um valor inteiro $n < 5000$ que corresponde ao número de elementos que aparecem na segunda linha. A segunda linha contém n valores inteiros, separados entre si por um espaço.

Saída

A saída é formada por uma linha contendo os n valores lidos.

Exemplo

Entrada					
5					
1	2	3	4	5	
Saída					
1	2	3	4	5	

20 Os Verdadeiros Sete Anões da Branca de Neve



(++++)

Todos os dias, enquanto os anões estão ocupados nas minas, Branca de Neve prepara o jantar para eles: sete cadeiras, sete pratos, sete garfos e sete facas para sete anões famintos. Um dia, em vez de sete, nove anões voltaram das minas (ninguém sabe como ou por quê). Cada um deles afirma ser um dos sete anões da Branca de Neve. Felizmente, cada anão usa uma touca com um número inteiro positivo (menor que 100) escrito nela. Branca de Neve, uma matemática famosa, já havia observado, há muito tempo, que a soma dos números nas toucas de seus sete anões era exatamente 100. Escreva um programa que determina quais anões são legítimos, ou seja, escolhe sete dos nove números que totalizem 100.

Entrada

A entrada conterá um inteiro T , o número de casos de testes, e, para cada caso de teste, nove linhas de entrada. Cada uma com um inteiro entre 1 e 99 (inclusive). Todos os números serão distintos.

Saída

A saída deve conter, para cada caso de teste, exatamente sete linhas. Cada uma com um dos números nas toucas dos anões de Branca de Neve (em ordem crescente).

Exemplo

Entrada
2
7
8
10
13
15
19
20
23
25
8
6
5
1
37
30
28
22
36
Saída
7
8
10
13
19
20
23
1
5
6
8
22
28
30

21 Soma Elementos do Vetor



(+)

Elabore um programa que armazene n valores em um vetor, e depois imprima a soma de todos os elementos inseridos.

Entrada

A entrada contém duas linhas. A primeira, contém um valor inteiro $n < 5000$ que corresponde ao número de elementos que aparecem na segunda linha. A segunda linha contém n valores inteiros, separados entre si por um espaço.

Saída

A saída é formada por uma linha contendo um valor inteiro correspondente à soma dos n elementos no vetor.

Exemplo

Entrada
8
1 5 6 3 6 7 4 3
Saída
35