

Lista de vetores em sala de aula

Prof. Dr. Gustavo Teodoro Laureano

Profa. Dra. Luciana Berretta

Prof. Dr. Thierson Rosa Couto

Sumário

1	Achei	2
2	Aula Cancelada	3
3	Contagem	5
4	Inverte Vetor	6
5	Maior Elemento	7

1 Achei



(+)

Faça um programa que receba um vetor V com N números inteiros e posteriormente receba M números e verifique se eles estão ou não no vetor.

Entrada

O programa terá apenas um caso de teste. Na primeira linha do caso de teste há um número inteiro N , $1 \leq N \leq 1000$, representando o tamanho do vetor V . Na linha seguinte haverá N números inteiros separados por um espaço em branco, que são os N valores do vetor V . Na terceira linha será informado um número inteiro M , $1 \leq M \leq 1000$, representando a quantidade de buscas que serão efetuadas no vetor. Logo em seguida haverá M linhas, cada uma com um número inteiro que deve ser buscado no vetor V .

Saída

Seu programa gera M linhas de saída. Cada uma com o resultado da Busca dos M números inteiros no vetor V . Quando o valor estiver no vetor V escreva “ACHEI”, quando não estiver escreva “NAO ACHEI”, com todas as letras maiúsculas e sem acentos. Ao final quebre uma linha.

Exemplo

Entrada
10
9 0 1 3 8 2 7 4 6 5
4
1
23
4
7
Saída
ACHEI
NAO ACHEI
ACHEI
ACHEI

2 Aula Cancelada



(+++)

Um professor X tem uma turma de N alunos. Frustrado com a falta de disciplina, ele decide cancelar a aula se menos de K alunos estão presentes quando a aula começa. Dado o tempo de chegada de cada aluno, determinar se a aula é cancelada. Caso a aula não seja cancelada, imprima uma lista com os alunos que chegaram antes do início da aula em ordem contrária à mostrada na entrada.

Entrada

A primeira linha apresenta dois números inteiros separados por um espaço: N (alunos da turma) e K (mínimo de presenças para que a aula não seja cancelada), com $0 \leq N, K, \leq 1000$. Na segunda linha há N inteiros separados por espaços (A_1, A_2, \dots, A_n) descrevendo os tempos de chegada para cada aluno. Suponha que esta ordem seja a mesma da lista de presença do professor, com o primeiro aluno descrito na entrada sendo o aluno 1 e assim por diante. Nota: horários de chegada não-positivos ($A_i \leq 0$) indicam que o aluno chegou cedo ou na hora; horários de chegada positivos ($A_i > 0$) indicam o aluno chegou A_i minutos tarde.

Saída

O programa apresenta uma mensagem com a palavra “SIM” se a aula é cancelada, e “NAO” caso contrário. Após imprimir a mensagem quebre uma linha. Se a aula não for cancelada, imprima os M alunos presentes antes do início da aula (ou seja, com $A_i \leq 0$) na ordem contrária da lista de entrada.

Exemplo

Entrada
4 3
-1 -3 4 2
Saída
SIM

Entrada
4 2
0 -1 2 1
Saída
NAO
2
1

Entrada
10 10
0 0 0 0 0 0 0 0 0 1
Saída
SIM

Entrada
2 1
-8 -4
Saída
NAO
2
1

Entrada
2 1
1 2
Saída
SIM

Entrada
10 4
-93 -86 49 -62 -90 -63 40 72 11 67
Saída
NAO
6
5
4
2
1

Entrada
10 10
23 -35 -2 58 -67 -56 -42 -73 -19 37
Saída
SIM

Entrada										
10	1									
88	-17	-96	43	83	99	25	90	-39	86	
Saída										
NAO										
9										
3										
2										

3 Contagem



(+)

Dado um vetor V de tamanho N e um inteiro K , contabilize quantos elementos de V são maiores ou iguais ao inteiro K .

Entrada

O programa terá apenas um caso de teste. O programa deve ler, obrigatoriamente, um número N que pertença ao intervalo $1 \leq N \leq 1000$. Se N lido não for válido, o programa deve fazer uma nova leitura de N . Caso N seja válido, N representa o tamanho do vetor V . Na próxima linha há N números inteiros separados por um espaço em branco cada, representando cada elemento do vetor V . E finalmente, na última linha há um inteiro K .

Saída

Seu programa gera apenas uma linha de saída contendo um número inteiro representando quantos elementos do vetor V são maiores ou iguais ao inteiro K . Após a impressão do valor quebre uma linha.

Exemplo

Entrada	
0	
-3	
4	
1 2 3 4	
0	
Saída	
4	

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

Entrada	
10	
1 2 3 4 5 6 7 8 9 10	
20	
Saída	
0	

Entrada	
1	
2	
3	
Saída	
0	

Entrada	
4	
1 4 6 4	
4	
Saída	
3	

4 Inverte Vetor



(++)

Faça um programa que receba um vetor V de N inteiros e construa um vetor W com os mesmos elementos de V, porém invertidos, mostre os vetores V e W e o maior elemento de V e o menor elemento de W.

Entrada

A entrada contém apenas um caso de teste com 2 linhas. Na primeira linha há um inteiro N, $1 < N \leq 1000$, representando o tamanho do vetor V. Na segunda linha há N valores inteiros separados por um espaço em branco cada, que são os valores do vetor V.

Saída

O programa deve gerar 4 linhas de saída. A primeira linha deve haver N inteiros separados por um espaço em branco cada, representando os elementos do vetor V. Atenção, após o último elemento de V não deve haver um espaço em branco. A segunda linha deve haver N inteiros separados por um espaço em branco cada, representando os elementos do vetor W. Atenção, após o último elemento de V não deve haver um espaço em branco. A terceira linha deve haver apenas um inteiro, representando o maior elemento de V. A quarta linha deve haver apenas um inteiro, representando o menor elemento de W. Após imprimir a quarta linha da saída, quebre uma linha.

Exemplo

Entrada
5
7 8 4 9 2
Saída
7 8 4 9 2
2 9 4 8 7
9
2

Entrada
8
235 6 23 5 78 123 89 4
Saída
235 6 23 5 78 123 89 4
4 89 123 78 5 23 6 235
235
4

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

5 Maior Elemento



(+)

Faça um programa que receba vários vetores e informe para cada um deles qual o maior elemento e o índice (da primeira ocorrência) em que encontra-se tal elemento.

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 N linhas de saída, com dois inteiros separados por um espaço em branco. O primeiro inteiro é o índice da primeira ocorrência do maior elemento do vetor e o segundo inteiro é o maior valor do vetor. Após a impressão de cada saída, inclusive a última, quebre uma linha.

Exemplo

Entrada
10
6 54 7 3 73 6 67 23 6 9
5
9 8 7 6 5
8
0 1 2 3 4 5 6 7
0
Saída
4 73
0 9
7 7