

AG22S - Algoritmos 2

Revisão C e Vetores

1ª Prática

Nome: _____ Turma: _____ Código: _____ Nº: _____

Número de Problemas Corretos	Nota
0	0.0
1	2.5
2	5.0
3	7.5
4+	10.0

1. Inicial

Faça um programa em C que leia um valor inteiro N que representa o tamanho do vetor. Após isso, defina um vetor de double de tamanho N . O programa deve fazer a leitura de N valores double e salva-los no vetor. Por fim, o programa deve imprimir a posição do vetor e o valor correspondente como mostrado no exemplo abaixo (com uma casa decimal) em ordem crescente e decrescente.

Entrada

Um valor inteiro N , seguido de N valores double.

Saída

Os valores do vetor em ordem crescente e decrescente conforme exemplo abaixo.

Exemplo de entrada	Exemplo de saída
5 10.0 5.0 9.0 8.0 7.0	V[0] = 10.0 V[1] = 5.0 V[2] = 9.0 V[3] = 8.0 V[4] = 7.0 V[4] = 7.0 V[3] = 8.0 V[2] = 9.0 V[1] = 5.0 V[0] = 10.0

2. Par

Faça um programa em C que leia um valor inteiro N que representa o tamanho do vetor. Após isso, defina um vetor de inteiros de tamanho N . O programa deve fazer a leitura de N valores e salva-los no vetor. Por fim, o programa deve imprimir a posição do vetor e o valor somente dos números pares do vetor.

Entrada

Um valor inteiro N , seguido de N valores inteiros.

Saída

A posição e os valores pares do vetor.

Exemplo de entrada	Exemplo de saída
5 10 8 1 4 7	V[0] = 10 V[1] = 8 V[3] = 4
3 3 5 7	

3. Menor que X

Faça um programa em C que leia um valor inteiro N que representa o tamanho do vetor. Após isso, defina um vetor de float de tamanho N . O programa deve fazer a leitura de N valores e salva-los no vetor. Em seguida, o programa deve ler um valor float X . Por fim, o programa deve mostra somente os valores menores que X (com duas casas decimais) e a posição do vetor em que elas estão.

Entrada

Um valor inteiro N , seguido de N valores float e, ao final, um float X .

Saída

Os valores menores que X do vetor e a sua respectiva posição no vetor.

Exemplo de entrada	Exemplo de saída
5 6.3 14.9 8.3 3.15 5.5 8.3	V[0] = 6.30 V[3] = 3.15 V[4] = 5.50

4. Exponenciação de vetores

Faça um código que leia dois vetores, $V = \{v_1, v_2, \dots, v_N\}$ e $W = \{w_1, w_2, \dots, w_N\}$, de inteiros longos, ambos de tamanho N da entrada padrão (ver formato em Entrada) e imprima o somatório, S , da exponenciação entre os valores que estão na mesma posição dos vetores, de modo que os valores do vetor V correspondam a base e os valores do vetor W correspondam ao expoente.

Formalmente:

$$S = \sum_{i=1}^N (v_i^{w_i}) = v_1^{w_1} + v_2^{w_2} + \dots + v_N^{w_N}$$

Entrada

O caso de teste é composto por três linhas. A primeira linha contém um inteiro $1 \leq N \leq 30$, indicando o tamanho dos vetores. A segunda linha contém N inteiros separados por um espaço em branco com os valores do vetor V . A terceira linha contém N inteiros em ordem separados por um espaço em branco com os valores do vetor W .

Saída

Imprima uma linha contendo o valor S .

Exemplo de entrada	Exemplo de saída
5 1 2 3 4 5 2 2 2 2 1	35

5. Busca Sequencial

Faça um programa em C que encontre a primeira posição de valores em um vetor. Primeiramente leia um vetor, V de tamanho N , da entrada padrão no formato especificado abaixo. Após isso, para cada uma das M consultas retorne a primeira posição do vetor na qual tal valor pode ser encontrado.

Entrada

- (a) A entrada inicia com uma linha contendo um único inteiro $0 < N \leq 20$ indicando a quantidade de itens do vetor V .
- (b) A segunda linha contém N inteiros, sendo que o primeiro deve ser armazenado na posição $V[0]$, o segundo na posição $V[1]$ e assim sucessivamente.
- (c) A terceira linha contém o um único inteiro $0 < M \leq 20$ indicando o número de consultas.
- (d) Finalmente, a quarta linha contém M inteiros, cada um representando um valor a ser consultado.

Saída

Para cada consulta imprima uma linha com a posição do vetor na qual o valor sendo consultada se encontra OU a mensagem "NOT FOUND" se o valor não pode ser encontrada.

Exemplo de entrada	Exemplo de saída
3 2 7 1 5 2 1 7 8 2	0 2 1 NOT FOUND 0
5 4 3 1 2 0 2 0 1	4 2