

Problema 1 - Soma de Vetores

Quando temos dois vetores de mesma dimensão $\vec{u} = (u_0, u_1, u_2, \dots, u_n)$ e $\vec{v} = (v_0, v_1, v_2, \dots, v_n)$, a soma $\vec{u} + \vec{v}$ é dada por:

$$\vec{u} + \vec{v} = (u_0 + v_0, u_1 + v_1, u_2 + v_2, \dots, u_n + v_n)$$

Partindo do código q2inicial.c, preencha as funções não implementadas e na função main:

- ler do usuário um inteiro **n1**, um inteiro **n2**
- alocar dinamicamente um vetor **u** de **n1** números inteiros e outro **v** de **n2** números inteiros usando a função `alocarVetor`
- ler **n1** inteiros para **u** e **n2** inteiros para **v**
- chamar a função `somaVetores`
- escrever na tela o vetor resultante da soma dos dois vetores ou ‘dimensoes incompativeis’ caso $n1 \neq n2$

Exemplos

Input	Output
3 4 1 3 4 3 4 2 4	dimensoes incompativeis
3 3 3 -2 1 4 -2 5	7 -4 6

Problema 2 - Ocorrências no vetor

No código exemplo12.c (parte 4) vimos como criar uma função que retorna indiretamente o maior número de um vetor de inteiros e o respectivo índice. Agora você deverá fazer algo parecido. A função deve retornar o endereço base de um vetor contendo todos os índices de um determinado número buscado no vetor. Você também deve acrescentar um retorno indireto através de ponteiro para armazenar a quantidade de ocorrências. A função deve seguir a seguinte assinatura:

```
int * buscaNoVetor(int *v, int n, int valor, int *maior, int *qtd)
```

onde **v** é o endereço base do vetor de inteiros a ser analisado, **n** a quantidade de inteiros nesse vetor, **valor** representa o valor buscado no vetor e **qtd** um ponteiro para inteiro que contém o endereço onde será armazenado a quantidade de ocorrências desse inteiro buscado. A função deve retornar NULL caso não encontre ocorrências.

A função main deve ler um inteiro **n** que representa a quantidade de elementos do vetor de inteiros. O vetor deve ser alocado dinamicamente. Em seguida, o programa deve ler do usuário os **n** inteiros e um valor inteiro a ser buscado. Depois o programa deve obter os índices de todas as ocorrências desse valor informado. O programa deve escrever na saída a quantidade de ocorrências e os índices. Lembre-se de liberar todos os vetores alocados dinamicamente.

Exemplos

Input	Output
5 2 5 2 2 1 2	3 0 2 3
3 1 2 3 7	Nenhuma ocorrencia
7 1 1 1 1 1 0 1 1	6 0 1 2 3 4 6

Problema 3 - Sopa de letrinhas

Crie uma função **misturar** que recebe 2 strings como parâmetros e retorna uma nova string (alocada dinamicamente) alternando as letras de ambas as strings. Caso não possuam o mesmo tamanho, as letras restantes da string maior devem ser colocadas ao final da nova string. Por exemplo, ao misturar “teste” e “algo”, obtém-se “taelsgtoe”. Já ao misturar “asa” e “inconstitucionalidade” obtém-se “aisnaconstitucionalidade”.

```
char * misturar(char *str1, char *str2)
```

A função main deve ler duas strings (uma por linha, com ou sem espaços) de até 100 caracteres, chamar a função misturar e escrever na tela o resultado. Não se esqueça de liberar a memória do que foi alocado dinamicamente.