

(\*) 0. Escreva um programa que gera  $N$  valores aleatórios do tipo `int` no intervalo  $[20, 80]$ , onde  $N$  é uma constante dada. O programa deve calcular e imprimir a média (não se preocupe com o arredondamento), e depois mostrar cada um dos valores, dizendo se ele é maior, menor ou igual à média. Por fim, o programa deve mostrar quantos dos valores eram maiores e quantos eram menores do que a média.

---

(\*) 1. Escreva um programa que gera  $N$  valores aleatórios do tipo `float` no intervalo  $[-40, +40]$ , onde  $N$  é uma constante dada. O programa deve calcular e imprimir a média dos positivos, e depois mostrar cada um dos valores positivos, dizendo se o valor é maior, menor ou igual à média. Já os valores negativos não devem ser armazenados em um vetor.

---

(\*) 2.

a) Escreva um programa que lê  $N$  valores do tipo `int`, onde  $N$  é uma constante dada. O programa deve imprimir os valores lidos na ordem inversa à fornecida. Você não deve usar  $N$  variáveis para armazenar os valores!

b) Modifique o programa para que o usuário forneça o valor de  $N$ , sabendo que  $N \leq 1024$ .

---

(\*\*) 3. Suponha que dois números formam um par de Foolano se ambos têm os mesmos dígitos, na mesma quantidade, desconsiderando-se os zeros à esquerda. Por exemplo, os pares abaixo são pares de Foolano:

1234 e 4321; 1122 e 1212; 101 e 101; 344 e 443.

Já os pares abaixo não são pares de Foolano:

123 e 124; 221 e 112; 022 e 220.

a) Escreva uma função que retorna 1 se 2 números dados formam um par de Foolano, ou 0 do contrário. O protótipo da função deve ser:

```
int ehParDeFoolano (unsigned int n1, unsigned int n2);
```

(\*\*\*\*) OPCIONAL, faça depois que terminar a lista inteira! b) Escreva uma versão da função `ehParDeFoolano` que não faça uso de vetores. Você pode criar para isso uma função auxiliar que recebe como parâmetro um número e um dígito e retorne o número de ocorrências daquele dígito no número. Existe alguma vantagem em resolver este problema usando vetores?

(\*\*\*) 4. (Adaptado da OBI 2012)

Tombólia do Oeste e Tombólia do Leste travaram uma guerra durante 50 anos. O motivo da guerra era o tamanho do território de cada país. Pelo bem da população dos dois países, os governos resolveram fazer um tratado para finalizar a guerra.

O tratado consiste em fazer uma divisão justa, e certamente contínua, do território. Eles resolveram pedir sua ajuda para calcular o ponto de divisão do território. Depois de tantos anos de guerra, os países não podem lhe pagar uma viagem para ver previamente o território que será dividido. Ao invés disso, eles prepararam uma lista  $a_1, a_2, \dots, a_N$  de inteiros que indicam o tamanho de cada seção do território. A seção  $a_1$  é vizinha da seção  $a_2$  que por sua vez é vizinha da seção  $a_3$ ; e assim por diante. Os governos querem uma divisão em uma seção  $k$  de tal forma que  $a_1 + a_2 + \dots + a_k = a_{k+1} + a_{k+2} + \dots + a_N$ .

Sua tarefa é, dada uma lista de inteiros positivos  $a_1, a_2, \dots, a_N$ , determinar a seção  $k$  tal que soma dos comprimentos das seções  $a_1$  até  $a_k$  é igual à soma dos comprimentos das seções  $a_{k+1}$  até  $a_N$ .

Entrada:

O programa recebe primeiro um inteiro  $2 \leq N \leq 100$ , que indica o número de seções do território. Ele então deve ler  $N$  inteiros positivos que indicam os comprimentos das seções.

Saída:

Seu programa deve imprimir uma única linha contendo um inteiro que indica a seção do território onde acontecerá a divisão.

Nota: é garantido que sempre existe uma divisão que satisfaz as condições dos países.

Exemplos:

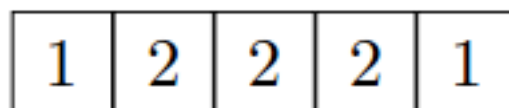
Entrada	Saída
4 5 3 2 10	3
9 2 8 2 8 4 4 4 4 4	4

(\*\*\*) 5. (Adaptado da OBI 2012)

Leonardo é um garoto fascinado por jogos de tabuleiro. Nas férias de janeiro, ele aprendeu um jogo chamado Campo Minado, que é jogado em um tabuleiro com  $N$  células dispostas na horizontal. O objetivo desse jogo é determinar, para cada célula do tabuleiro, o número de minas explosivas nos arredores da mesma (que são a própria célula e as células imediatamente vizinhas à direita e à esquerda, caso essas existam). Por exemplo, a figura abaixo ilustra uma possível configuração de um tabuleiro com 5 células:



A primeira célula não possui nenhuma mina explosiva, mas é vizinha de uma célula que possui uma mina explosiva. Nos arredores da segunda célula temos duas minas, e o mesmo acontece para a terceira e quarta células; a quinta célula só tem uma mina explosiva em seus arredores. A próxima figura ilustra a resposta para esse caso.



Leonardo sabe que você participa da OBI e resolveu lhe pedir para escrever um programa de computador que, dado um tabuleiro, imprima o número de minas na vizinhança de cada posição. Assim, ele poderá conferir as centenas de tabuleiros que resolveu durante as férias.

Entrada:

O programa recebe primeiro um inteiro  $2 \leq N \leq 50$ , que indica o número de células no tabuleiro. Ele então deve ler  $N$  inteiros com valor 0 ou 1, que indicam se existe (1) ou não (0) uma mina em cada posição do tabuleiro, lidos na sequência em que aparecem no tabuleiro.

Saída:

O programa deve mostrar  $N$  valores, indicando o número de minas explosivas nos arredores de cada casa do tabuleiro, na sequência em que aparecem no tabuleiro.

Exemplos:

Entrada	Saída
5 0 1 1 0 1	1 2 2 2 1
5 0 1 1 1 0	1 2 3 2 1

PS: Será que você realmente precisa de vetores para resolver este problema?