



Universidade Federal do Ceará

Curso de Engenharia da Computação - Prof. Danilo Alves

1ª LISTA – PROGRAMAÇÃO COMPUTACIONAL

1. (1,5 pts) Na calçada em frente ao Palácio Imperial, não se sabe a razão, existe uma sequência de N números desenhados no chão. A sequência é composta apenas pelos números de 1 a N . Veja um exemplo na coluna (a) da figura abaixo, para $N=12$.

3	3
2	2
5	5
2	2
10	10
4	4
4	4
7	7
12	12
2	2
8	8
10	10
(a)	(b)

Ninguém sabe o significado da sequência e, justamente por isso, várias teorias malucas surgiram. Uma delas diz que a sequência representa, na verdade, apenas um valor que estaria relacionado a um grande segredo dos imperadores. Esse valor é a quantidade máxima de números da sequência que poderiam ser marcados com um círculo, de modo que a sequência de números marcados não contenha dois números iguais consecutivos e seja composta de no máximo dois números distintos.

A coluna (b) da figura ilustra uma sequência de 4 números marcados que obedece a restrição acima. Você consegue verificar que essa é, de fato, a quantidade máxima possível de números numa sequência marcada?

Neste problema, dada a sequência original de números desenhados no chão da calçada, seu programa deve computar e imprimir a quantidade máxima de números da sequência que poderiam ser marcados com um círculo sem que haja dois números iguais consecutivos na sequência marcada e tal que ela seja composta de no máximo dois números distintos.

Entrada

A primeira linha da entrada contém um inteiro N representando o tamanho da sequência. As N linhas seguintes contém, cada uma, um inteiro V_i , para $1 \leq i \leq N$, definindo a sequência de números desenhados no chão da calçada imperial.

Saída

Seu programa deve imprimir uma linha contendo um número inteiro representando a quantidade máxima de números da sequência que poderiam ser marcados com um círculo sem que haja dois números iguais consecutivos na sequência marcada e tal que ela seja composta de no máximo dois números distintos.

Restrições

$$1 \leq N \leq 500$$

$$1 \leq V_i \leq N, \text{ para } 1 \leq i \leq N$$

Exemplos

Entrada	Saída
1 1	1
12 3 2 5 2 10 4 4 7 12 2 8 10	4

2. (2 pts) Roberto tem um conjunto de lápis com 10 tons diferentes de uma mesma cor, numerados de 0 a 9. Numa folha de caderno quadriculado alguns quadrados foram coloridos inicialmente com o tom 0. Roberto precisa determinar, para cada quadrado Q não colorido, qual é a distância dele para o quadrado mais próximo de tom 0. A distância entre dois quadrados é definida com o número mínimo de movimentos ortogonais (para: \it esquerda, direita, cima, baixo) para ir de um quadrado para o outro. O quadrado Q, então, deve ser colorido com o tom cuja numeração corresponde à distância determinada. Se a distância for maior ou igual a 9, o quadrado deve ser colorido com o tom 9. Seu programa deve colorir e imprimir a folha quadriculada dada na entrada.

Entrada

A primeira linha da entrada contém apenas um inteiro N, determinando as dimensões da folha quadriculada, N x N. As N linhas seguintes definem a folha inicialmente. Cada linha contém uma sequência de N caracteres: `*' se o quadrado não está colorido, e `0" se está colorido com o tom 0.

Saída

Seu programa deve imprimir o tabuleiro totalmente colorido, de acordo com a regra definida acima.

Restrições

$$3 \leq N \leq 1000.$$

Exemplos

Entrada	Saída
8 **000*** ***** *****0** ***** *****000 *****0 0*****0 *****	21000123 32111123 43221012 34332111 23321000 12332110 01233210 12344321
Entrada	Saída
3 *** *** **0	432 321 210

3. (1,5 pt) Uma professora de programação, cansada de que os estudantes cheguem tarde, decidiu que vai cancelar a aula se há poucos presentes.

Ela representa a entrada dos estudantes como um array de tempos de chegada tarde, em minutos. Por exemplo, se um estudante chegou 10 minutos atrasado, outro 5 minutos antes da hora, outro com 3 minutos de atraso, e outro pontual, poderá representar assim:

```
int alunosDaSegunda[4] = {10, -5, 3, 0};
```

Com essa informação e a quantidade mínima de estudantes para que suceda o curso, a professora quer saber se a aula acontecerá. Por exemplo, supondo que a quantidade mínima de estudantes para que a aula aconteça é de 2 alunos, então o curso da segunda-feira se realizará, porque houve um estudante que foi pontual e um estudante que chegou cedo.

```
acontece(alunosDaSegunda, 2) // Recebe o array com os tempo e a quantidade mínima de  
aluno para ter aula.
```

```
1 //Retorno da função, equivalente a true
```

Mas se a quantidade mínima fosse 3, a aula não aconteceria:

```
acontece(alunosDaSegunda, 3)

0 //Retorno da função, equivalente a false
```

Seu programa deve conter as seguintes funções:

1. *acontece*, que diz se a aula sucederá de acordo com o array dos estudantes que entraram.
2. *aberturas*, que utiliza um array com os arrays dos estudantes que entraram nos outros dias, e a quantidade mínima de estudantes, e diga quais os dias em que as aulas aconteceram e quais não, retornando um array com 1 para os dias que tiveram aulas e 0 para os que não tiveram. Na saída de seu programa apresente esse array. Por exemplo:

```
aberturas([alunosDaSegunda, alunosDaTerça, alunosDaQuarta], 2)

[1, 0, 0]
```

4. (1 pt) As árvores utópicas crescem de uma forma particular, em dois ciclos:
- cada primavera dobram seu tamanho
 - cada verão crescem um metro

Se Laura planta uma árvore utópica com um metro, no final do outono, qual seria sua altura depois de N ciclos?

Alguns exemplos:

si $N = 0$, sua altura será 1 metro (não cresceu nada)

si $N = 1$, sua altura será de 2 metros (dobrou a altura na primavera)

si $N = 2$, sua altura será de 3 metros (cresceu um metro mais no verão)

si $N = 3$, sua altura será de 6 metros (dobrou a altura na primavera seguinte)

E assim ...

Escreva a função `alturaArvoreUtopica`, que utilize uma quantidade de ciclos de crescimento, e retorne a altura resultante da árvore de Laura.

5. (1 pt) Dados dois strings (um contendo uma frase e outro contendo uma palavra), determine o número de vezes que a palavra ocorre na frase.

Exemplo:

Para a palavra ANA e a frase:

ANA E MARIANA GOSTAM DE BANANA

O seu programa deve retornar para o usuário que a palavra ocorre 4 vezes na frase.

6. (1,5 pt) Podemos calcular o seno de um número segundo a série de Taylor-Maclaurin:

$$\text{Sen}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + \dots$$

Faça um programa em C que lê um valor para x e calcule o valor de Sen(x). O valor deve ser calculado utilizando funções e enquanto o termo $\frac{x^n}{n!}$ calculado for maior que 10E-12.

7. (1,5 pts) Utilizando números primos, crie um programa para cada item abaixo:
- Um programa que recebe um número do usuário e informa se é primo.
 - Um programa que recebe um número do usuário e retorna a sequência de números primos até o número informado.
 - Um programa que recebe um número do usuário e retorna a soma dos números primos existentes até o número informado.