



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA

# **Maratona de Programação**

## **Seletiva UFPE 2016**

**2ª Sessão de Warm-up**  
**Folha de questões**

Maratona de Programação CIn-UFPE  
**24 de Fevereiro de 2016**

## **A - Saltando primos**

Professor Jaime adora tudo relacionado a números primos. Lembre-se que um número primo é um número positivo maior do que um e somente divisível por 1 e por si mesmo. Ele está trabalhando agora em uma propriedade de um conjunto de números primos chamado de campeão de salto. Um número inteiro  $N$  é chamado de "campeão salto" se é a diferença mais frequente entre primos consecutivos.

Por exemplo, considere os números primos consecutivos 2 3 5 7 11. As diferenças entre os números primos são 1 2 2 4. Portanto, para este conjunto de números primos, o campeão de salto é exatamente 2 (que ocorre duas vezes).

Ele realmente gostaria de saber para qualquer conjunto de números primos que é o seu campeão salto correspondente. Você pode ajudá-lo?

### **Entrada:**

A entrada é uma linha com dois números inteiros  $L$  e  $U$  ( $0 < L < 1000$ ), separados por um espaço único, que representam os limites inferior e superior (respectivamente) a considerar.

### **Saída:**

Um número, representando o campeão de salto, seguido de uma quebra de linha. Caso não haja campeão de salto, imprima "-1", sem as aspas, também seguido de uma quebra de linha. É garantido que, se houver um campeão, ele é unicamente determinado.

### **Exemplo 1:**

Entrada:	Saída:
2 3	1

### **Exemplo 2:**

Entrada:	Saída:
2 11	2

## **B - Quantos números são maiores ou menores que uma variação da média?**

Serão dados  $n$  números correspondendo as notas da turma de alunos de uma escola. Escreva um programa que leia essas notas e calcule quantas estão 10% acima da média e quantas estão 10% abaixo da média.

### **Entrada:**

Na primeira linha você receberá um número inteiro  $n$ , onde  $n \leq 20000$ . Em seguida você receberá  $n$  números reais, um em cada linha, correspondendo as notas de cada aluno da turma.

### **Saída:**

Imprima na primeira linha da saída a média das notas formatada com duas casas decimais. Na segunda linha imprima quantas notas ficaram 10% acima da média. Na terceira linha imprima quantas notas ficaram 10% abaixo da média.

### **Exemplo:**

Entrada:	Saída:
5 10 10 9 1 0	6.00 3 2

## **C - Intersecção entre listas**

Faça um programa que recebe números de dois arrays inteiros com 20 posições cada e apresente a intersecção dos arrays ordenado de forma crescente. Lembrando que a intersecção são os elementos repetidos em ambos os arrays, mas sem repetição (cada número pode aparecer uma única vez no resultado).

### **Entrada:**

Você receberá 40 números inteiros, um por linha, sendo 20 correspondendo ao primeiro array e 20 ao segundo.

### **Saída:**

Você deve imprimir a intersecção entre os dois arrays, um número por linha, ordenado de forma crescente. Caso a intersecção seja vazia, imprima a string "VAZIO".

### **Exemplo:**

Entrada:	Saída:
1	10
2	11
3	12
4	13
5	14
6	15
7	16
8	17
9	18
10	19
11	20
12	
13	
14	
15	
16	
17	
18	
19	
20	
10	
11	
12	
13	
14	
15	
16	
17	

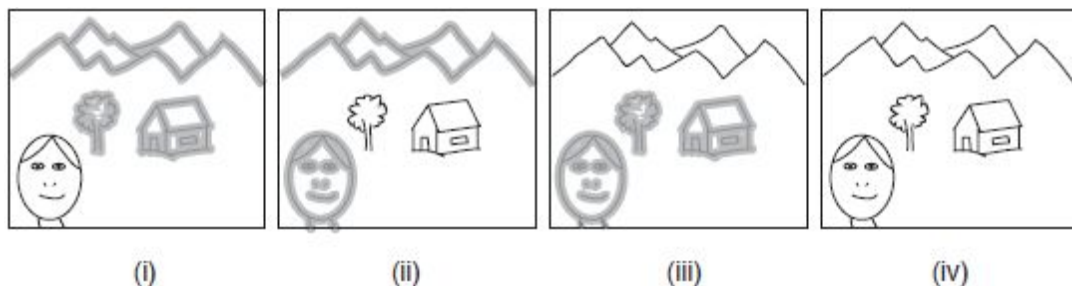
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		

## **D - Foco nas fotografias**

Daniel está fazendo um curso de Visão Computacional e decidiu reproduzir um trabalho interessante que ele viu na aula: ele tirou algumas fotos de uma mesma cena, variando apenas o foco, para depois combiná-las em apenas uma foto, em que todos os objetos da cena fiquem focados juntos. Para isso, ele necessita que cada objeto apareça focado em pelo menos uma fotografia.

Por exemplo, no quadro a seguir, (i), (ii) e (iii) são 3 fotos da mesma cena, cada uma tirada com um foco diferente, (iv) é a imagem combinada gerada por Daniel. Perceba nessas fotos, que a pessoa está em um plano focal, a árvore e a casa estão em outro plano e as montanhas estão em outro plano, ou seja, foram 3 planos focais no total.

Mas Daniel sabe que, para cada objeto, há um intervalo fechado dos planos focais em que esse objeto fica claramente visível. Ou seja, o objeto ficará claramente visível em um intervalo de planos e não apenas em um plano.



Como o cartão de memória da câmera de Daniel tem uma pequena capacidade, ele pediu a sua ajuda. Tendo em conta os intervalos de foco de todos os objetos na cena que pretende fotografar, determinar o número mínimo de fotos que ele deve tomar para que cada objeto fique focado em pelo menos uma das fotos.

**Entrada:**

A entrada consiste em vários testes de casos. A primeira linha de cada caso de teste contém um inteiro  $N$  ( $1 \leq N \leq 50$ ) que indica o número de objetos na cena. Cada uma das  $N$  linhas seguintes contém dois números inteiros,  $A$  e  $B$  ( $1 \leq A \leq B \leq 50$ ), que indica os extremos do intervalo de foco de cada objeto.

**Saída:**

Para cada caso de teste, você deve imprimir uma linha com um número inteiro que indica o número mínimo de fotos que Daniel tem que realizar, seguida de uma quebra de linha.

**Exemplo:**

Entrada:	Saída:
3	2

1 3 2 5 4 6	
-------------------	--

## **E - Array Hash**

Você terá como uma entrada várias linhas, cada uma com uma string. O valor de cada caractere é computado como segue:

$$\text{Valor} = (\text{Posição no alfabeto}) + (\text{Elemento de entrada}) + (\text{Posição do elemento})$$

Todas posições são baseadas em zero. 'A' tem posição 0 no alfabeto, 'B' tem posição 1 no alfabeto, ... O cálculo de hash retornado é a soma de todos os caracteres da entrada. Por exemplo, se a entrada for:

CBA  
DDD

então cada caractere deverá ser computado como segue:

2 = 2 + 0 + 0 : 'C' no elemento 0 posição 0  
2 = 1 + 0 + 1 : 'B' no elemento 0 posição 1  
2 = 0 + 0 + 2 : 'A' no elemento 0 posição 2  
4 = 3 + 1 + 0 : 'D' no elemento 1 posição 0  
5 = 3 + 1 + 1 : 'D' no elemento 1 posição 1  
6 = 3 + 1 + 2 : 'D' no elemento 1 posição 2

O cálculo final de hash será 2+2+2+4+5+6 = 21.

### **Entrada:**

A entrada contém vários casos de teste. A primeira linha de entrada contém um inteiro N que indica a quantidade de casos de teste. Cada caso de teste inicia com um inteiro L ( $1 \leq L \leq 100$ ) que indica a quantidade de linhas que vem a seguir. Cada uma destas L linhas contém uma string com até 50 letras maiúsculas ('A' - 'Z').

### **Saída:**

Para cada caso de teste imprima o valor de hash que é calculado conforme o exemplo apresentado acima.

### **Exemplo:**

Entrada:	Saída:
5	21
2	25
CBA	30
DDD	4290
1	295
Z	
6	



A

B

C

D

E

F

6

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

1

ZZZZZZZZZZ

## **F - Produto simples**

Leia valores inteiros. A seguir, calcule o produto entre estes valores e atribua esta operação à variável PROD. A seguir mostre a variável PROD com mensagem correspondente.

Entrada:

O arquivo de entrada contém  $n$  valores inteiros. Você não receberá um número indicando quantos inteiros você receberá. Portanto, você deverá ler os números até não ter mais nenhum número na entrada.

Saída:

Imprima a variável PROD conforme exemplo abaixo, com um espaço em branco antes e depois da igualdade. Não esqueça o fim de linha após o produto.

Exemplo:

Entrada:	Saída:
2 2	Prod = 4

## **G - Produto dos dígitos**

Para um número inteiro  $N$  dado, não-negativo, encontrar o mínimo  $Q$  natural tal que o produto de todos os dígitos de  $Q$  é igual  $N$ .

Entrada:

A entrada contém um número inteiro positivo  $N$ , tal que  $9 < N < 100000$ .

Saída:

O menor número, o qual o produto dos seus dígitos é igual a  $N$ , e "-1", se não existir, seguido de uma quebra de linha.

Exemplo:

Entrada:	Saída:
10	25