



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação
Disciplina Fundamentos de Programação**

AP1 1º semestre de 2017

IMPORTANTE

- Prova sem consulta e sem uso de qualquer aparato eletrônico.
- Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
- Você pode usar lápis para responder as questões.
- Ao final da prova, devolva as folhas de questões e as de respostas.
- Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

1ª Questão (2,5 pontos)

Faça um programa que leia do usuário números inteiros, um por linha, até que um número negativo seja digitado. Caso algum número tenha sido informado, escreva os dois maiores números lidos.

Entrada

A entrada é composta por uma ou mais linhas contendo números inteiros. A última linha sempre conterá um número negativo, o delimitador de término das entradas.

Saída

Caso apenas o número delimitador seja lido escreva: "Nenhum valor válido!!!";

Caso apenas um número não negativo seja lido, antes do delimitado, escreva: "Apenas um valor foi lido: " seguido do respectivo valor;

Caso dois ou mais valores forem lidos, escreva: "Os dois maiores números lidos foram: ", seguido do maior valor lido, seguido da string " e ", seguido do segundo maior valor lido.

Exemplos

Entrada	Saída
-5	Nenhum valor válido!!!

Entrada	Saída
13 -8	Apenas um valor foi lido: 13

Entrada	Saída
3 2 13 18 5 -2	Os dois maiores números lidos foram: 18 e 13

2ª Questão (2,5 pontos)

Faça um programa que peça ao usuário as dimensões de uma matriz bidimensional, chamada de `valores`, de números inteiros, a ser gerada aleatoriamente no intervalo 10 a 99. Via subprogramação:

- (1) Mostre o conteúdo da matriz gerada (0,5 ponto);
- (2) Identifique e escreva o conteúdo da linha de maior soma, caso haja empate escreva uma delas (1,0 ponto);
- (3) Identifique e escreva o conteúdo da coluna de maior soma, caso haja empate escreva uma delas (1,0 ponto).

Dica

Utilize a função `random.randint(a, b)`, disponível na API, que retorna um número randômico inteiro entre `a` e `b`, inclusive.

Entrada

A entrada é composta de apenas uma linha, que define a quantidade `L` de linhas e a quantidade `C` de colunas da matriz a ser gerada.

Saída

Inicialmente: `L` linha(s), onde cada linha possui `C` valor(es) inteiro(s) no intervalo 10 a 99, seguida(s) de uma linha em branco;

O conteúdo da linha de maior soma, composta de `C` valor(es) inteiro(s) no intervalo 10 a 99, seguida de uma linha em branco;

O conteúdo da coluna de maior soma, composta de `L` valor(res), escrito(s) um por linha;

Ao final, escrever linha em branco.

Exemplo

Entrada	Saída
5 6	13 28 45 50 26 10 27 24 22 33 88 11 90 25 85 23 76 55 77 15 31 29 13 14 66 41 50 20 47 11 90 25 85 23 76 55 13 27 90 77 66

3ª Questão (2,5 pontos)

A sequência de Padovan é uma sequência de números naturais $P(n)$ definida pelos valores iniciais

$$P(0) = P(1) = P(2) = 1$$

e a seguinte relação recursiva

$$P(n) = P(n - 2) + P(n - 3) \text{ se } n > 2.$$

Alguns valores da sequência são: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28...

Faça um programa que implemente a função recursiva `padovan(n)`, que recebe um número inteiro n e retorna o n -ésimo termo da sequência de Padovan. Seu programa receberá várias entradas. Para cada entrada, seu programa ativará a função `padovan(n)` e escreverá o termo correspondente retornado pela função.

Entrada

A entrada é composta por várias linhas. Cada linha a ser processada contém um número natural $n \geq 0$. A última linha contém um valor negativo que indica o término da sequência de números naturais. Essa linha não deverá ser processada.

Saída

Para cada linha a ser processada seu programa deverá imprimir uma linha contendo o termo correspondente na sequência de Padovan.

Exemplo

Entrada	Saída
7	5
13	28
0	1
16	65
-1	

4ª Questão (2,5 pontos)

Conforme visto em aula, o algoritmo de busca binária recebe um vetor ordenado e um valor a ser procurado nesse vetor. A cada passo, o algoritmo divide o espaço de busca em dois grupos até encontrar o elemento que está sendo procurado ou até o espaço de busca se tornar vazio.

Escreva um programa que implemente e ative cada um dos subprogramas definidos a seguir:

- (1) `lerVetor()` – Essa função lê um conjunto ordenado de `n` strings informados via entrada padrão (teclado) e o armazena no vetor retornado por ela. (valor 0,3 ponto)
- (2) `lerValor()` – Essa função lê da entrada padrão (teclado) uma string a ser procurada e a retorna. (valor 0,2 ponto)
- (3) `buscaBinaria(lista, valor)` – Essa função realiza a busca binária de `valor` no conjunto ordenado `lista`. A função retorna a chave (índice) do valor caso esse tenha sido encontrado ou `-1` caso esse não tenha sido encontrado na lista de strings. Seguindo o padrão de indexação adotado pela linguagem Python, o primeiro elemento na lista possui índice 0, o segundo possui índice 1 e assim por diante. (valor 1,5 pontos)
- (4) Ao final, o programa principal deve imprimir na saída padrão (vídeo) o texto "O valor foi encontrado na posição ", seguido pelo índice retornado pela função `buscaBinaria`, ou o texto "Valor não encontrado" caso a função `buscaBinaria` retorne `-1`. (valor 0,5 ponto)

Entrada

A primeira linha da entrada contém um valor inteiro `N`, indicando quantos strings compõe a coleção ordenada. As `N` linhas seguintes contém as strings da coleção (uma por linha). A linha que sucede a coleção armazena a string a ser procurada.

Saída

O programa deve imprimir uma única linha. Caso o valor buscado tenha sido encontrado, essa linha deve conter o texto "O valor foi encontrado na posição ", seguido pelo índice da posição do valor na coleção de strings. Caso o valor buscado não seja encontrado, essa linha deve conter o texto "Valor não encontrado".

Exemplo

Entrada	Saída
6 Bon Jovi Bruce Springsteen Madonna Michael Jackson Mick Jagger Paul McCartney Mick Jagger	O valor foi encontrado na posição 4

Boa Avaliação!