

Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina Fundamentos de Programação

AP1 1º semestre de 2019

IMPORTANTE

- Serão aceitos apenas soluções escritas na linguagem Python 3.
 - Prova sem consulta e sem uso de qualquer aparato eletrônico.
 - Use caneta para preencher o seu nome e assinar nas folhas de questões e 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 (2,0 pontos)

Utilizando subprogramação, faça um programa que leia linhas da entrada padrão até que uma linha vazia seja informada. Com exceção da linha vazia, todas as demais linhas contêm um número inteiro, de 0 até 9, de um candidato a uma eleição. Seu programa deve escrever quais foram os números dos candidatos digitados e quantas vezes cada um foi votado, no formato exemplificado a seguir. Caso nenhum candidato seja digitado então escreva a mensagem: "Nenhum voto ocorreu!!!".

Exemplos

Entrada	Saída
<code><linha deixada em branco></code>	Nenhum voto ocorreu!!!

Entrada	Saída
9 <code><linha deixada em branco></code>	Candidato 9 obteve 1 voto(s)

Entrada	Saída
4 7 2 5 4 2 5 5 8 <code><linha deixada em branco></code>	Candidato 2 obteve 2 voto(s) Candidato 4 obteve 2 voto(s) Candidato 5 obteve 3 voto(s) Candidato 7 obteve 1 voto(s) Candidato 8 obteve 1 voto(s)

Distribuição de Pontos

Entrada – 0,2 pontos; Processamento – 1,6 pontos; Saída – 0,2 pontos.

2ª Questão (3,0 pontos)

Utilizando subprogramação, faça um programa que leia linhas da entrada padrão até que uma linha vazia seja informada. Com exceção da linha vazia, todas as demais devem conter os valores inteiros, separados por espaços em branco, de cada linha da matriz numérica a ser construída. Mostre o conteúdo da matriz lida, obedecendo o formato apresentado nos exemplos a seguir. Escreva também qual é a linha com maior soma de valores e qual é o total da soma nesta linha, caso haja empate escreva uma delas. Caso a primeira linha seja vazia, escreva a mensagem: "Matriz vazia, não existe linha com maior soma!!!!!!".

Exemplos

Entrada
<i><linha deixada em branco></i>
Saída
----- Matriz Lida ----- ----- Matriz vazia, não existe linha com maior soma!!!

Entrada
132 232 342 4 222 234 1232 13 <i><linha deixada em branco></i>
Saída
----- Matriz Lida ----- 132 232 342 4 1232 13 222 234 ----- Linha com maior soma foi a linha com valor(es): [1232, 13], cuja soma = 124

Entrada
8 5 6 7 13 -4 5 55 2 3 300 2 10 2 4 8 <i><linha deixada em branco></i>

Saída			
----- Matriz Lida -----			
8	5	6	7
13	-4	5	55
2	3	300	2
10	2	4	8

Linha com maior soma foi a linha com valor(es): [2, 3, 300, 2], cuja soma = 307			

Distribuição de Pontos

Entrada (com conversão de tipos) – 0,5 pontos; Processamento – 2,0 pontos; Formatação e impressão da saída – 0,5 pontos.

3ª Questão (3,0 pontos)

Escreva um programa que receba como entrada uma linha contendo um número inteiro positivo N, seguido por N linhas compostas por caracteres 'x'. Após a leitura da entrada, seu programa deverá ordenar as linhas pelo método da seleção (*Selection Sort*), visto em aula. A ordenação deverá ser em ordem crescente pelo número de caracteres contidos na linha. Ou seja, linhas com menor quantidade de caracteres antecedem linhas com maior quantidade de caracteres. Ao final, o programa deverá imprimir as linhas ordenadas.

Exemplos

Entrada	Saída
1 xx	xx

Entrada	Saída
3 xxx xx x	x xx xxx

Entrada	Saída
5 xxxxx x xx xxxxxxxx xxxxxxxx	x xx xxxxx xxxxxxxx xxxxxxxx

Distribuição de Pontos

Leitura da entrada – 0,3 pontos; *Selection Sort* – 2,5 pontos; Saída – 0,2 pontos.

Distribuição de Pontos

Obs. 1: A implementação de outros algoritmos de ordenação resultará em pontuação parcial de, no máximo, 1,5 pontos.

Obs. 2: É proibido o uso de funções de ordenação presentes na API do Python. A pontuação atribuída a essa questão será igual a zero caso tais funções sejam utilizadas.

4ª Questão (2,0 pontos)

Uma situação que todo programador enfrenta, por mais experiente que seja, é lidar com erros de sintaxe e problemas que levem a erros execução introduzidos por engano no código fonte. Nessa questão você deverá analisar o código fornecido e identificar os erros contidos nele. Para cada erro, indique claramente uma solução para sua correção.

```
def gcd(a, b):  
    if b == 0:  
        return a  
    else:  
        return gcd(b, a % b)  
  
x = input("Informe x, um valor inteiro não negativo: ")  
while x < 0  
    x = input("Valor inválido. Informe outro valor para x: ")  
  
y = input("Informe y, um valor inteiro não negativo: ")  
while y < 0  
    y = input("Valor inválido. Informe outro valor para y: ")  
  
r = gcd(x, y)  
print(O máximo divisor comum de %d e %d é %d. % (a, b, r))
```

Distribuição de Pontos

A distribuição de pontos não é informada para evitar a dedução de quantos erros foram introduzidos no código fornecido.

Boa Avaliação!