Laboratório de Programação Competitiva

Profa. Silvia Brandão

2024.2

Aula de hoje

- Resolução de exercícios em Python
- Discussão sobre estratégias de resolução de código.

Exercício

Elabore funções em Python para:

- ler um conjunto de números reais, diferentes de zero, armazenando-os no vetor **vet**, até que o usuário digite zero. Zero não poderá fazer parte do vetor;
- calcular a média aritmética dos valores dos vetor;
- determinar o maior valor do vetor;
- determinar a quantidade de números ímpares no vetor

Observação: Um número real só pode ser classificado como par ou ímpar se for um número inteiro. Então converta o número real em inteiro.

Para testar as funções (métodos) , implemente o programa principal com a chamada das funções e imprima os resultados obtidos em saída formatada.

Exercícios em Python – Beecrowd 2782

beecrowd | 2782

Escadinha

Por OBI Brasil

Timelimit: 1

Dizemos que uma sequência de números é uma escadinha, se a diferença entre números consecutivos é sempre a mesma. Por exemplo, "2, 3, 4, 5" e "10, 7, 4" são escadinhas. Note que qualquer sequência com apenas um ou dois números também é uma escadinha! Neste problema estamos procurando escadinhas em uma sequência maior de números. Dada uma sequência de números, queremos determinar quantas escadinhas existem. Mas só estamos interessados em escadinhas tão longas quanto possível. Por isso, se uma escadinha é um pedaço de outra, consideramos somente a maior. Por exemplo, na sequência "1, 1, 1, 3, 5, 4, 8, 12" temos 4 escadinhas diferentes: "1, 1, 1", "1, 3, 5", "5, 4" e "4, 8, 12".

Entrada

A primeira linha da entra contém um inteiro **N** ($1 \leq N \leq 1000$) indicando o tamanho da sequência de números. A segunda linha contém **N** inteiros definindo a sequência. O valor dos números da sequência está entre -10^6 e 10^6 inclusive.

Saída

Imprima uma linha contendo um inteiro representando quantas escadinhas existem na sequência.

Exemplos de Entrada	Exemplos de Saída
8 1 1 1 3 5 4 8 12	4
1 112	1
5 11 -106 -223 -340 -457	1

				10	7	4	
			5	10			
2	3	4	5				

					4	8	12
				5	4		
		1	3	5			
1	1	1					

```
def contar_escadinhas(seq):
    # Se houver apenas um número, há exatamente uma escadinha
    if len(seq) == 1:
        return 1

    # Inicialização
    num_escadinhas = 0
    j = 0
    n = len(seq) – 1

    # Percorrendo a sequência para encontrar escadinhas
    while j < n:
        # Identifica a diferença inicial
        i = j
        diff = seq[i+1] - seq[i]

        # Avança até que a diferença mude
        while i < n and seq[i+1] - seq[i] == diff:
            print(f"seq[{i}] = {seq[i]} e seq[{i+1}] = {seq[i+1]} e diff = {diff}")
            i += 1

        # Quando a escadinha termina, contamos
        num_escadinhas += 1
        j = i

    return num_escadinhas
```

```
# Leitura da entrada
N = int(input())
seq = list(map(int, input().split()))

# Processamento e saída do resultado
print(contar_escadinhas(seq))
```

					4	8	12
				5	4		
		1	3	5			
1	1	1					

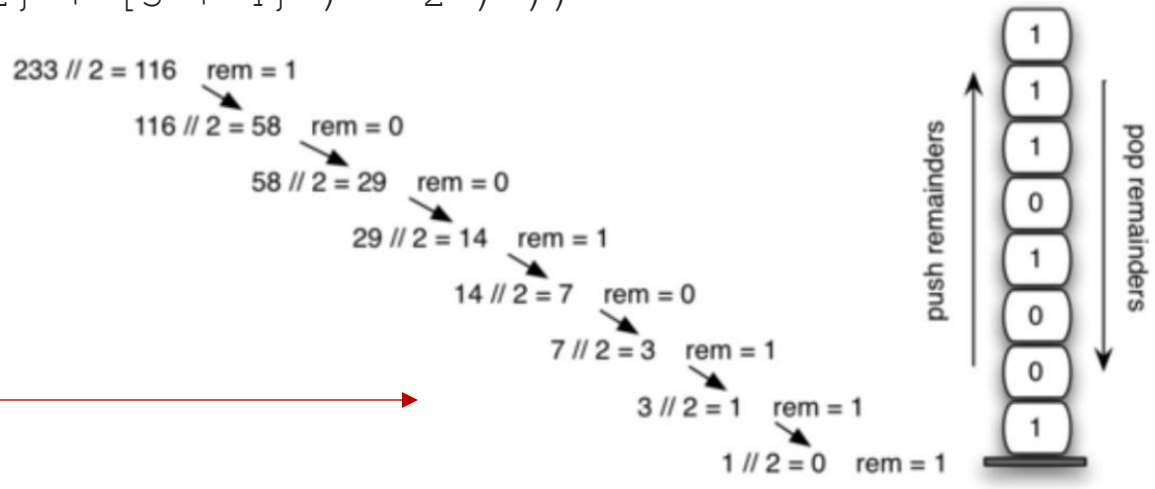
				10	7	4
			5	10		
2	3	4	5			

Tarefa: usando o TAD Pilha – entregar

1. Faça um código em Python que use a estrutura de dados pilha para verificar o balanceamento dos parênteses, colchetes e chaves em expressões aritméticas:

```
print('Expressoes simples')
print(verifica_expressao_simples('((1 + 2) * (3 + 4)) - (5 + 6)'))
print()
print(verifica_expressao_simples('((1 + 2) * (3 + 4))))'))
print()
print(verifica_expressao_simples('(((((1 + 2) * (3 + 4))'))
print()
print('Expressoes compostas')
print(verifica_expressao_composta('{ [ (1 + 2) + (3 + 4) ] * 2 }'))
print()
print(verifica_expressao_composta('{ [ (1 + 2} + [3 + 4} ) * 2 )'))
```

2. Implemente uma função em Python que converta um número inteiro arbitrário na base 10 (decimal) para a representação binária.



Operações com listas e Fatiamento de listas

As operações a seguir são suportadas pela maioria dos tipos de sequência, mutáveis e imutáveis.

Sejam s e t são sequências do mesmo tipo, n , i , j e k são inteiros e x é um objeto arbitrário que atende a qualquer tipo e restrições de valor impostas por s .

x in s	True se um item de s é igual a x
x not in s	False se um item de s é igual a x
s + t	Concatenação de s e t
s*n ou n*s	Equivalente a adicionar s a si mesmo n vezes
s[i]	Elemento na posição i de s
s[i:j]	Fatia s de i para j
s[i:j:k]	Fatia s de i para j com o passo k
len(s)	Comprimento de s
min(s)	Menor item de s
max(s)	Maior item de s
sum(s)	Soma dos elementos de s
s.count(x)	Número total de ocorrências de x em s

https://docs.python.org/3.6/library/stdtypes.html#mutable-sequence-types

Podemos verificar se existe um elemento especifico na lista e ver a quantidade de elementos com:

```
valores = [0, 1, 2, 3]
print(len(valores))
print(4 in valores)
print(valores + valores)
print(valores[1:])
print(valores[:-1])
```

A saída será algo como:

```
4
False
[0, 1, 2, 3, 0, 1, 2, 3]
[1, 2, 3]
[0, 1, 2]
```

Exercícios - entregar próxima aula

1. Faça:
   - Mostre-me as seguinte listas, derivadas de:
     - [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
       - Intervalo de 1 a 9
       - Intervalo de 8 a 13
       - Números pares
       - Números ímpares
       - Todos os múltiplos de 2, 3 e 4
       - Lista reversa
       - Razão entre a soma do intervalo de 10 a 15 pelo intervalo de 3 a 9 em float!

2. Elaborar um programa em Python que recebe uma lista de tuplas contendo o nome e a nota de cada aluno. Calcule a média das notas dos alunos.

   EXEMPLO:

   alunos = [("Ana", 7.5), ("Bruno", 8.0), ("Carlos", 6.5), ("Diana", 9.0), ("Eduardo", 7.0)]

Exercícios - entregar próxima aula

3) Faça um programa que permita ao usuário digitar o seu nome seguido do sobrenome em uma mesma linha de entrada. Em seguida, mostre o sobrenome do usuário utilizando somente letras maiúsculas, concatenado com uma vírgula e um espaço em branco; em seguida, concatenar também com o nome, utilizando somente a primeira letra maiúscula e as demais minúsculas, seguido de um ponto '.'. Implementar um programa Python, usando uma lista, que interage sistematicamente com o usuário solicitando nomes para serem cadastrados (armazenados numa lista). A cada nome inserido o programa pergunta se deseja encerrar e quando o usuário responde 'sim', todos os nomes previamente armazenados deverão ser mostrados na tela, conforme mostra o exemplo: Ana Ferreira → FERREIRA, Ana.

Dica: lembre−se que ao informar o nome o usuário pode digitar letras maiúsculas ou minúsculas.

4) Gerar uma sequência de números inteiro pares, usando o range() e armazenar em uma matriz quadrada N a raiz quadrada de cada valor gerado. Imprima a matriz.
5) Faça um programa que leia um número indeterminado de valores, correspondentes a notas, e os armazene em uma lista. Encerrando a entrada de dados quando for informado um valor igual a -1 (que não deve ser armazenado). Após esta entrada de dados, faça:
   a. Mostre a quantidade de valores que foram lidos;
   b. Exiba todos os valores na ordem em que foram informados, um ao lado do outro;
   c. Exiba todos os valores na ordem inversa à que foram informados, um abaixo do outro;
   d. Calcule e mostre a soma dos valores;

   Sugestão: use as funções e operadores de lista

Solução dos exercícios

https://colab.research.google.com/drive/1i4fvZ81nJWc5_YAqemeah8qM7_tjsWIH?usp=sharing

Próxima Aula

Teoria dos Números x Teoria dos Números em Python:

- Algoritmo de Euclides. MDC. Implementação.
- Cálculo do Fatorial tendo como resultado números grandes.
- Teorema fundamental da aritmética: MDC, MMC, Primos, Fatorial. Implementações.
- Discussão em grupo sobre estratégias de resolução.

Não se esqueçam do Uniube+