Prova 2: Computação I 2016.2

Prof.ª Laura Moraes

19 de Dezembro de 2016

Observações:

- Leia com atenção os enunciados até o final antes de começar a escrever as respostas.
- Boas práticas de programação também são parte da avaliação. Lembre-se de usar nomes significativos para variáveis e funções, organizar seu código com funções, colocar comentários e identar corretamente.

1. Dada a função:

```
def misterio(v):
   for i in range(len(v)-1,0,-1):
       position = 0
       for location in range(1,i+1):
           if v[location]>v[position]:
               position = location

       temp = v[i]
       v[i] = v[position]
       v[position] = temp

    return v
```

   (a) (0.75 ponto) O que ela faz?

   (b) (0.75 ponto) Qual seria o resultado para a entrada [2, 256, 48, 64, 32, 128]

   (c) (1 ponto) Reescreva a função trocando o segundo **for** por uma chamada **while**.

2. (a) (1 ponto) Descreva o que são **dicionários** e como eles se diferenciam de tuplas e listas.

   (b) (1.5 pontos) Existem 36 combinações possíveis ao se jogar dois dados. Um simples par de loops consegue enumerar todas as combinações. No entanto, a soma dessas combinações é mais interessante de se analisar. Crie um **dicionário** utilizando **a soma dos dois dados como chave**. O valor deverá ser **quantas vezes essa soma aparece**.

3. (2.5 pontos) Considerando a função e as chamadas no *python shell* abaixo, marque a opção correta de saída.

```
def charada(A, B):
    if len(A[0]) == len(B):
        C = []
        for i in range(len(A)):
            linha = []
            for j in range(len(B[0])):
                valor = 0
                for x in range(len(A[0])):
                    valor += A[i][x] * B[x][j]
                linha.append(valor)
            C = C + [linha]
        return C
    else:
        print "Erro!"
```

```
>>> charada([[1,2],[4,5]],[[1,2],[4,5]])
>>> charada([[1,0],[0,1]],[[20,75],[10,5]])
```

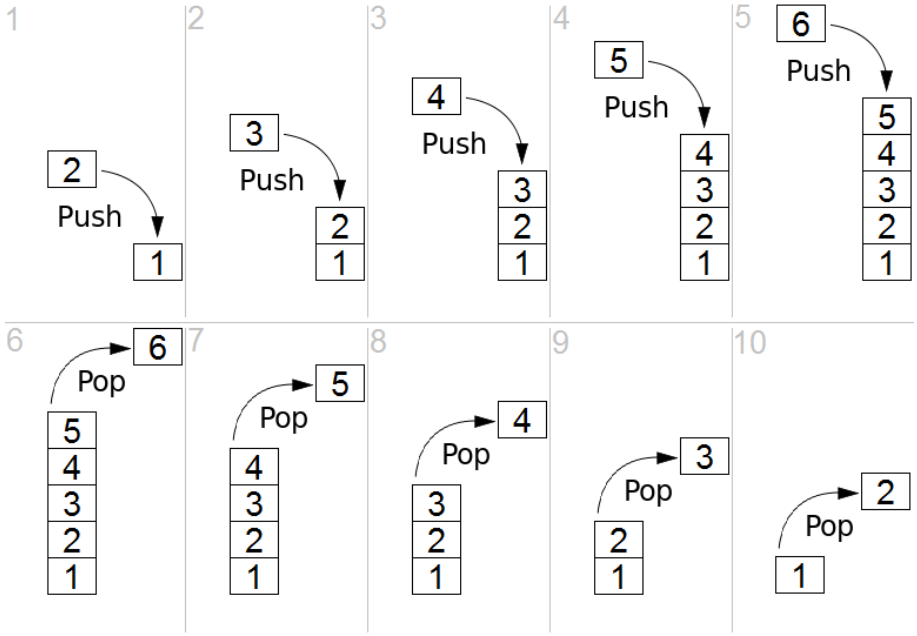
(A) [[1, 4], [8, 10]] e [[20, 10], [75, 5]]

(B) [[9, 12], [24, 33]] e [[20, 75], [10, 5]]

(C) [[9, 12], [24, 33]] e [[75, 20], [5, 10]]

(D) [[5, 14], [14, 41]] e [[20, 10], [75, 5]]

4. Em ciência da computação, LIFO (acrônimo para a expressão inglesa *Last In, First Out* que, em português significa último a entrar, primeiro a sair) refere-se a estruturas de dados do tipo pilha. Usa-se os termos *push* e *pop* para denominar a inserção e remoção de elementos da pilha, respectivamente. Usa-se o termo *top* para consultar o elemento do topo da pilha, sem removê-lo. Veja a figura abaixo para entender o funcionamento da pilha:



Pilhas podem ser facilmente implementadas em Python utilizando listas. A primeira posição da lista é a parte inferior da pilha e, conforme elementos são adicionados à pilha, adiciona-se elementos à lista. A última posição da lista representa o topo da pilha.

Implemente um programa que dê ao usuário **3 opções**: *push*, *pop* e *top*. Se o usuário escolher *push*, o programa deverá **pedir um elemento e adicioná-lo no topo da pilha** (última posição da lista). Se a opção *pop* for escolhida, o programa deve **exibir ao usuário qual elemento se encontra no topo da lista e removê-lo**. Por fim, se a opção *top* for escolhida, o programa **exibe ao usuário o valor no topo da pilha**, mas não remove-o da lista.