Untitled

January 20, 2020

1 Computer Science G11

1.1 Term 2 Test 3

Date Mon Jan 20 2020

1.2 NAME:

2 1 (KtiCa) (40%) Implement a stack class. Make sure to satisfy the following requirements and definitions.

In computer science, a stack is an abstract data type (for the sake of this problem, think of this as a class) that serves as a collection of elements, with two principal operations: push, which adds an element to the collection, and. pop, which removes the most recently added element that was not yet removed.

The interface of your code should be as follows

```
s = Stack() #creates an empty stack
            # returns None
s.pop()
s.push('hello')
s.push(3)
s.len()
            # returns 2, i.e, the number of items in the stack
            # returns 3 removing it from the stack
s.pop()
s.len()
            # returns 1
s.push([5,7,9,'foo'])
s.len()
            # retutns 5
s.pop()
            # returns 'foo'
s.pop()
            # returns 9
```

3 2 (KtiCa) (40%) Implement a *joining* of stacks, that is, an operator that takes two stacks and returns a new stack containing the union of the both stacks' elements.

The interface of your code should be as follows

```
A = Stack([0,2,4,6])
B = Stack([1,3,5,7])
C = A*B
C.len() # 8
C.pop() # 7
C.pop() # 5
```

4 3 (KtiCA) (20%) We may say that two stacks are equal when they store the same elements. Implement the comparision of stacks.

The interface of your code should satisfy

```
s = Stack()
A = Stack([0, 'hello',s]
B = Stack()
B.push(0)
B.push('hello')
B.push(s)
A == B  # True
B.pop(s)
s.push(3)
B.push(s)
A == B  # False
```

[]: