Problem Set 2

Justin Ely

615.202.81.FA15 Data Structures

15 September, 2015

1a)

Set the variable i equal to the second element from the top of the stack. Leave the stack unchanged.

```
s = Stack()
a = s.pop()
b = s.pop()
i = b

s.push(b)
s.push(a)
```

1b)

Given an integer n, set the variable i equal to the nth element from the top of the stack. Leave the stack without its top n elements

```
s = Stack()

j = 1
while j < n:
    s.pop()
i = s.pop()
```

2a)

Set the variable i equal to the bottom element of the stack. Leave the stack unchanged.

```
s1 = stack()
s2 = stack()
```

```
while not s1.empty():
    s2.push(s1.pop())
i = s2.peek()

while not s2.empty():
    s1.push(s2.pop())
```

2b)

Set the variable i equal to the third element from the bottom of the stack..

```
s1 = stack()
s2 = stack()

while not s1.empty():
    s2.push(s1.pop())

s2.pop()
s2.pop()
i = s2.pop()
```

3a)

Iteration	Stack
0	{
1	{ [
2	{
3	{ [
4	{ [(
5	{ [
6	{

3b)

Iteration	Stack
0	(
1	((
2	(
3	({
4	({ (
5	({ (
6	({ ([
7	({ ([
8	({ (
9	({
10	(
11	

4)

```
def check_mirror(string):
    s = Stack()
    second_half = False

    for letter in string:
        if not second_half:
            s.push(letter)

        if letter == 'c':
            s.pop()
            second_half = True

        if s.empty():
            return False
        if letter == s.pop():
            return False

    return True
```

5)

```
# using function definition from problem 4
```

```
def pattern_match(string):
    for letter in string:
        s.push(letter)

        if letter == 'D':
            s.pop()

            if not check_mirror():
```

6)

7)

8)

9)

9a]

Prefix:
Postfix: A B + C $ D E - F + * G -

9b]

Prefix:
Postfix: A B C - D E - * F + G / $ H J - +

10)

10a]

10b]

10c]

$(A - B + C)^(D + E - F)$

10d]

11)

11a)

$((A + B) - C) - ((B + A)^C) = 0$

11b)

$(Ax(B+C))x(C+(B-A)) = 20$

12)