Polytechnic Tutoring Center

Exam 2 Review ANSWER KEY - CS 1114, Fall 2021

Disclaimer: This mock exam is only for practice. It was made by tutors in the Polytechnic Tutoring Center and is not representative of the actual exam given by the CS Department.

1. For each of the following, give the resultant output of the code. If the code results in an error, write ERROR. If the output is None, write NONE.

a.
```
my_tuple = (0, 1, 2)
my_tuple[2] = 4
print(my_tuple[2])                        ERROR
```

b.
```
my_str = ‘school’
print(my_str[5: :-2])                     loc
```

c.
```
cool_stuff = ‘Math is cool’
print(cool_stuff.split())                [‘Math’, ‘is’, ‘cool’]
```

d.
```
lst = [0, 1, 2, 3, 4]
lst.insert(2, 3)
print(lst)                                [0, 1, 3, 2, 3, 4]
```

e.
```
lst = [0, 1, 2, 3, 4]
print(lst.pop(2))                         2
```

f.
```
lst = [0, 1, 2, 3, 4]
lst.pop()
lst.append(5)
print(lst)                                [0, 1, 2, 3, 5]
```

g.
```
my_str = ‘The cow jumped over the moon’
print(my_str[2: -1])                      e cow jumped over the moo
```

h.
```
lst = [1, 2, 3, 4, 5, 6]
new_lst = lst.reverse()
print(new_lst[1])                         ERROR
```

2. a/ Write a program that prints out a triangle with specified height and character type. Call this function `display_triangle`. Assume that the height passed in is greater than 1.

For example:

```
>>> display_triangle(‘#’, 3)

  #
 ###
#####
```

```
def func(symbol, n):
     for i in range(n):
          spaces = “ “ * (n – i - 1)
          fill = symbol * (2 * i + 1)
          print(spaces + fill)
```

b/ Write a program that prints out a triangle of numbers with specified height and prints column-wise. For example, the first column will start with the specified range of values like (1,2,3,4 and 5), the second column will start from row number second, the third column will start from row number 3, and it will continue till the loop ends.

Call this function display_num_triangle.

For example:

```
>>> display_num_triangle(5)

1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

```
def display_num_triangle(n):
    for row in range(1, n+1):
        for column in range(row, 0, -1):
            print(column, end='')
        print()
```

3. What is the output of the following code?

```
def func1(x):
     for i in range(x, 0, -2):
          print(i**2, end = ‘ ‘)
     print()

def func2(x):
     while x > 1:
          x //= 2
          func1(x)

y = 10
func2(y)
```

```
25 9 1
4
1
```

4. Write a function that will identify if a given string is a palindrome. A palindrome is a word that when spelled backwards, reads the same as it does when spelled normally. An example of a palindrome would be the word “racecar.” The function should not distinguish between capital and lowercase letters.

```
def is_palindrome(s):
    front = 0
    back = len(s) – 1
    while (front < back):
        if s[front].lower() != s[back].lower():
            return False
        back -= 1
        front += 1
    return True
```

5. This question will be based entirely off the following code…

```
s = ‘the 167 20 lazy 210 brown 78 87 2 fox 245 123’
lst = s.split()
acc = 0
for elem in lst:
     acc += int(elem)
print(acc)
```

(a) Why does this code not work? **Can’t convert a string to an int**

(b) What type of error does it give? **ValueError**

(c) Can you fix it using a try/except block? Write the code to fix it.

Inside the for loop…

```
for elem in lst:
   try:
        acc += int(elem)
   except ValueError:
        print(“Value Error detected!!!”)

print(acc)
```

6. A file called scores.txt has a list of names and scores on each line in the following format…

```
Nick 12
Tina 20
Jack 8
Jill 18
...
```

(a) Write a function which reads this file and stores each (name, score) pair in a list, and stores all such tuples/lists in an aggregate list, returning this list. Be careful, if the scores.txt file does not exist, we don’t want to get an error, so you’ll have to find a way to get around this.

```
def read_file(file):
   res = []
   try:
         file_obj = open(file, ‘r’):
         for line in file_obj:
               line = line.strip()
               line = line.split()
               res.append(line)
         file_obj.close()

   except FileNotFoundError:
         print(“File Not Found”)

   return res
```

(b) Write a function which uses this list, and, given a person’s name, edits the score associated with that name.

```
def update_score(name, score, lst):
   for score in lst:
         if score[0] == name:
               score[1] = str(score)
```

(c) Write a function which writes to the same file, scores.txt, and then write the data from the aggregate lists into the file with the same format as the original, effectively updating the file. Again, be careful to not generate an error.

```
def write_file(file, lst):
     try:
          file_obj = open(file, ‘w’)
          for line in lst:
               s = line[0] + ’ ‘ + line[1] + ’\n’
               file_obj.write(s)
          file_obj.close()

     except FileNotFoundError:
          print(“File Not Found”)
```