



# Polytechnic Tutoring Center

## Exam 2 Review ANSWER KEY - CS 1113, 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. Determine the output of each of the following. If the code results in an error, write ERROR. If the output is None, write NONE.

- a/    `my_str = 'the quick brown fox'`                    **ERROR**  
      `my_str[0] = 'T'`  
      `print(my_str)`
- b/    `num1 = 4`    **ERROR**  
      `num2 = 3`  
      `print(num1 => num2)`
- c/    `my_str = 'The cow jumped over the moon'`  
      `print(my_str[2: -1])`                            **e cow jumped over the moo**
- d/    `my_str = 'school is cool'`                    **'s oh'**  
      `print(my_str[8:1:-2])`
- e/    `str1 = 'a'`                                        **'aaabbbbb'**  
      `str2 = 'b'`  
      `print(3*str1 + 5*str2)`
- f/    `my_str = 'RaceCar'`                            **'RACECAR'**  
      `print(my_str.upper())`
- g/    `str1 = 'abc'`                                    **TRUE**  
      `str2 = 'ABC'`  
      `print(str1 > str2)`
- h/    `str1 = 'abc'`                                    **FALSE**  
      `str2 = 'ABC'`  
      `print(str1 == str2)`

2. Write a function called `row_count` that, when given an integer, `n`, will produce `n` rows of numbers, each counting either up to or down from `n - i`, where `i`, represents the row number starting from the 0<sup>th</sup> row. When `i` is even, count up to `n`. When `i` is odd, count down from `n - i`. An example of the desired output is provided below:

```
>>> row_count(5)
```

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

```
def row_count(num):
    for i in range(num):
        if i % 2 == 0:
            for j in range(num - i):
                print(j + 1, end = ' ')
            print()
        else:
            for j in range(num - i):
                print(num - i - j, end = ' ')
            print()
```

3. 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()
```

4. Write a function that identifies whether the input string is a palindrome. A palindrome is a word that reads the same backwards as when read normally. For example, 'racecar' is a palindrome. Note that the algorithm should not distinguish between upper and lower case letters. For the purposes of this exercise, we will consider them to be the same. That is 'rAcEcar' would still be a palindrome.

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

5. What is the output of the following code?

```
def func(symbol, num):  
    for i in range(num):  
        spaces = " " * (num - i - 1)  
        fill = symbol * (2 * i + 1)  
        print(spaces + fill)  
  
func('*', 5)
```

**Output:**

```
    *  
   ***  
  *****  
 *****  
*****
```

6. What is the output of the following code?

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

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

num = 10
func2(num)
```

```
25 9 1
4
1
```

7. Write a function to return a string containing only different characters in lowercase from a given string (remove all duplicates).

```
aStr = 'aaaabbbb'  
print(removeDuplicate(aStr))  
>>>> 'ab'
```

```
aStr = 'aaaabbbbCCD'  
print(removeDuplicate(aStr))  
>>>> 'abcd'
```

```
aStr = ''  
print(removeDuplicate(aStr))  
>>>> ''
```

```
def removeDuplicate(aStr):  
    result = ''  
    for char in aStr:  
        char_lower = char.lower()  
        if char_lower not in result:  
            result += char_lower  
    return result
```

8. Given two variables **start** and **end**. Write a function to display all prime numbers  $p$  such as:  
**0** <= **start** <=  $p$  <= **end**

Print all these numbers on the same line separated by **ONE** space character

*~ 1 is not a prime number ~*

```
displayPrime(0,7)
```

```
>>>> 2 3 5 7
```

```
displayPrime(0,7)
```

```
>>>> 2 3 5 7
```

```
def displayPrime(start,end):
    for num in range(start,end+1):
        isPrime = True
        if num > 1:
            for i in range(2,num):
                if num % i == 0:
                    isPrime = False
            if isPrime:
                print(num, end = ' ')
```