

15-110 Refresher Session : Week 5 (Exam Review)

No Calculators, only Brains !!

1. THINK-PAIR-SHARE!!

(a) For loop Fun

Rules :

1. Use the number lines to trace the “for loop” for each turn by circling the values from start to stop
2. Your score will be the sum of all circled values
3. Write the python version of the for loop for each round
4. Best of 3 wins.

| | | | |
|---------|----------|--|----------------|
| ROUND 1 | Player 1 | For values of X from <u>1</u> to <u>18</u> step by <u>4</u> <small>starting value stopping value step</small> | SCORE _____ |
| | Player 2 | For values of X from <u>3</u> to <u>11</u> step by <u>2</u> <small>starting value stopping value step</small> | |
| ROUND 2 | Player 1 | For values of X from <u>3</u> to <u>17</u> step by <u>5</u> <small>starting value stopping value step</small> | SCORE _____ |
| | Player 2 | For values of X from <u>5</u> to <u>17</u> step by <u>3</u> <small>starting value stopping value step</small> | |
| ROUND 3 | Player 1 | For values of X from <u>6</u> to <u>11</u> step by <u>1</u> <small>starting value stopping value step</small> | SCORE _____ |
| | Player 2 | For values of X from <u>2</u> to <u>15</u> step by <u>6</u> <small>starting value stopping value step</small> | |

Reference : <https://studio.code.org/s/course4/lessons/8/levels/1>

(b) Spot the Error

In the given python function, the number is added to the sum if it is divisible by 2 and 3, or subtracted from the sum if it is greater than 4, or multiplied to the sum otherwise. Spot the errors and fix the code.

```
def error_1(n):
    for i in range(n) :
        if i % 2 == 0 and i % 3 == 0 :
            sum = sum + i
        else if i > 4 :
            sum = sum - i
        else :
            sum = sum * i
        return sum
```

(c) Food for thought:

When do you have to use else and when do you have elif in a conditional construct?

2. SuperCoder Activity

In the following questions, use a very short **python** code to solve the problem. Bonus points for the least number of lines.

1. Given a positive three digit integer n, find the sum of its first and last digits.

```
def sumFirstLast(n):
```

2. Given two numbers, return the integer quotient of the two if the first number is divisible by the second number or second number + 1; otherwise, return the product of the two numbers.

```
def twoNumbers(a ,b):
```

3. Given the radius of the circle, find its area if the square root of the radius is greater than 3, otherwise, return its circumference.

```
def areaCircum(r):
```

4. Given the area of a square tile as a and area of a square shaped room floor as A, return True if 4 tiles can fit into the room, and False otherwise.

```
def tiles(a, A):
```

5. There are n chocolates and s students. What is the maximum number of chocolates you can give the students without breaking or cutting them ?

```
def chocolates(n, s):
```

6. Given three numbers a, b, c; if they are strictly decreasing, return True if the greatest number is greater than the sum of the smaller 2, and false otherwise.

```
def decreasing(a, b, c):
```

3. Act like a computer:

Given the following function :

```
def ref_1(x,y) :
    if x > y :
        if x > 20 :
            return 20
        elif x > 10 :
            return 10
        if y > 5 :
            return 95
    else :
        return y
    return -1
```

What would the function call return for each of the functions below ?

1. `ref_1(8 , 7)`

2. `ref_1(7 , 8)`

3. `ref_1(5 , 4)`

4. **Act like a programmer:**

CMU-Q has a specific criteria in numbering courses. For example, all the Computer Science courses begin with 15, Math courses with 21, Physics with 33, etc. For the sake of this question, I am going to say that a course is difficult if the sum of the digits of the rest of the number (that is excluding 15 in 15110) is greater than 10 and easy otherwise.

Since this is a Computer Science course, your task is to write a python function `isDiffCSCourse(n)` which takes course number `n` of 5 digits and checks if it is a difficult level Computer Science course or not. The following output rules must be followed :

1. Prints "It is a difficult CS course" if it is a difficult course in CS
2. Prints "It is an easy CS course" if it is an easy course in CS
3. Prints "It is not a CS course" if it is not a CS course

Example :

```
isDiffCSCourse(15110) prints "It is an easy CS course"
isDiffCSCourse(15158) prints "It is a difficult CS course"
```

5. **Act like a computer:**

Given the following Python function

```
def ref_2(n):
    solution = 0
    for i in range(n-2) :
        if solution != 0 :
            solution = solution * i
    for index in range(1, n-2) :
        solution = solution + index
    for x in range(1, n + 3 , 3) :
        if solution % x == 0 and solution > x :
            solution = solution - x
    for anything in range(n , -2 , -1) :
        solution = solution + 1
    for that in range(n, -2) :
        solution = solution - 1
    return solution
```

What would the function call return for each of the functions below ?

1. `ref_2(5)`

2. `ref_2(10)`

6. **Act like a programmer:**

Write a python function `sumEvenSeven(limit)` which takes a number `limit` and returns the sum of all even multiples of 7 from 1 to the limit (inclusive). Example :

```
sumEvenSeven(7) returns 0 as there is no even multiple of 7 from 1-7
sumEvenSeven(14) returns 14 as 14 is the only even multiple of 7 from 1-14
sumEvenSeven(200) returns 1470
```

Challenge after you solve the problem : Try to solve it again without using conditionals

7. **Act like a computer:**

Given the following Python function

```
def ref_3(xyz, abc, pqr):
    x = xyz + abc * pqr
    y = (abc - xyz) % pqr
    z = (xyz // abc) / pqr
    if x % 4 == 0 and y % 2 == 0 or z < 0.9 and x * y > 4:
        return True
    else :
        return False
```

What would the function call return for each of the functions below ?

1. `ref_3(4 , 2 , 2)`

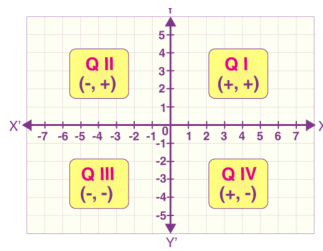
2. `ref_3(10 , 2 , 2)`

3. `ref_3(8 , 3 , 4)`

8. **Act like a programmer:**

A picture of a Cartesian plane quadrant is given below. Your task is to write a python function `quadrant(x, y)` which takes a x and y coordinates of a point as input and returns the quadrant number it belongs to. If (x,y) denotes the origin, return -1.

```
quadrant(1, 2) returns 1
quadrant(-1, 2) returns 2
quadrant(-1, -2) returns 3
quadrant(1, -2) returns 4
quadrant(0, 0) returns -1
```



9. **Act like a programmer:**

Write a python function `roots(a,b,c)` which takes three coefficients a, b, and c of a quadratic equation and returns the sum of the roots of the quadratic equation. If you have forgotten the quadratic formula, it is given below :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

10. **Challenging Question (Optional):**

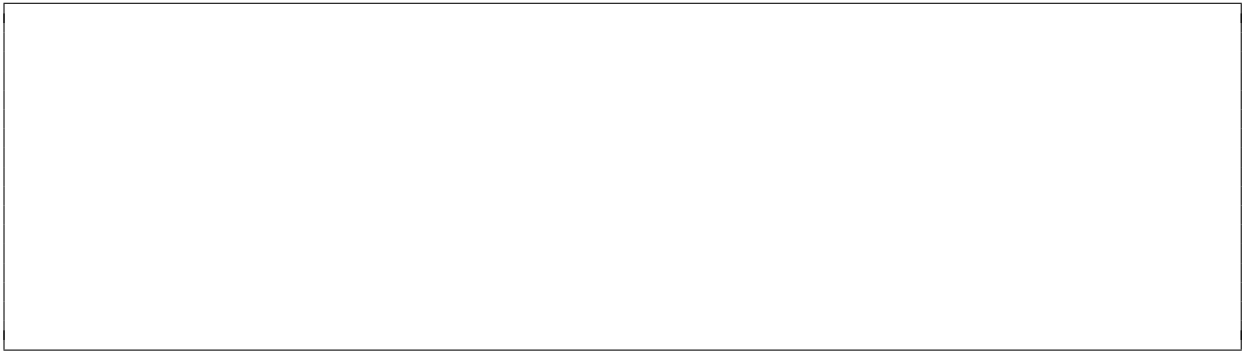
Jack's encrypter creates a code for all the numeric passwords that he has. The encrypter can only be used for numbers having 3 or 5 digits. The encrypter works the following way:

1. It takes the middle digit from the number
2. it adds the middle digit to all the digits of the number
3. It returns the created number

Your task is to create a function called `encrypt(n)` which takes a number and returns the code if it is valid, and -1 otherwise.

Example :

```
encrypt(90) returns -1
encrypt(111) returns 222
encrypt(10101) returns 21212
```



Challenge: Try to solve it with and without loops
