

Anirudh Bansal

All the best, give your honest attempt

## CSL 100: Midterm Exam

Name:

Roll Number:

**Section A: You can attempt any 4 out of 5 questions (5 marks each) in answer booklet.**

Please write a python program along with brief explanation of the variables used and the correctness of the program. If you forget some syntax, you can write pseudocode/flowchart with enough explanation. In case you can't write pseudocode, just write the steps of the algorithm in your own language (English/Hindi/Telugu/Tamil/Any other regional language).

**Q1.** Write a code to print **your name** in the centre of a rectangular box made of #. You can choose the dimensions of the box such that it looks symmetric. Example,

```
#####  
#                                     #  
#      Gagan Raj Gupta              #  
#                                     #  
#####
```

**Q2.** Print all numbers below 1000 whose sum of digits is 7.

**Q3.** You are given 3 lists of integers; you have to find their intersection. The lists can be of arbitrary size. Note that conversion to sets and using built-in intersection function is not allowed. Example below:

Input:

L1= [4,3,3,5,6,7]

L2= [2,3,4,6,7,9,11,3]

L3= [3,9,11,13,2,3,4,6,5]

Output:

[4,3,3,6]

**Q4.** Write a function min\_digit(list) to find the smallest number in a list. Now, use this function to sort the list in ascending order.

Input:

List = [4,3,3,5,7,6]

Output:

[3,3,4,5,6,7]

**Q5.** Please write a recursive function to convert a positive integer to its binary notation. In-built function to directly convert to binary is not allowed.

All the best, give your honest attempt

## Section B

Short answer questions (2 marks each). Please answer the question with a brief explanation in the answer booklet.

SQ1. What is the output of this function when the input by user is 10?

```
n = int(input())
sum_n = n * (n + 1) // 2 # Integer division
if sum_n % 2 == 0:
    print(n)
else:
    print(n - 1)
```

54

SQ2. Please write the missing code where the blanks are indicated in this recursive function to compute the factorial of a number.

```
def factorial (n):
    if (n==1):
        return _____
    else:
        return _____
```

SQ3. Please write the missing code where blanks are indicated in this swap function.

```
def swap_variables(a, b):
    a = a + b
    b = _____
    a = _____

    # Return the swapped values as a tuple
    return (a, b)
```

SQ4. What will be the output of the following program:

```
def addTenYearsToAge(currentAge):
    currentAge = currentAge + 10

realAge = 45
addTenYearsToAge(realAge)
print("Age in ten years " + str(realAge))
```

All the best, give your honest attempt

SQ5. Please write the missing code where blanks are indicated in this power function.

```
def power(base, exponent):  
    product = 1  
    counter = 1  
    while(counter <= exponent):  
        product = product * base  
        counter = counter + 1  
    return product
```

The expected behaviour of this function is that it raises the base to the power (exponent).  
For example the output of the following command will be 16.

```
print(power(2,4))
```

SQ6. Please write the missing code where blanks are indicated to solve the following problem: "If I leave my house at 6:52 am and run 1 mile at an easy pace (8 min 15 sec per mile), then 3 miles at faster pace (7 min 12 sec per mile) and 1 mile at easy pace again, what time do I get home for breakfast?"

```
easy_min = 8 + 15 / 60  
tempo_min = 7 + 12 / 60  
total_min = (tempo_min) * 3 + (easy_min) * 2  
print('Total minutes:', total_min)  
hours = _____  
minutes = _____  
print('Time in hours and minutes', hours, minutes)
```

SQ7. The following function is intended to check whether a string contains any lowercase letters, but there is some mistake. Can you spot the error and fix it?

```
def any_lowercase(s):  
    for c in s:  
        if c.islower():  
            return True  
    else:  
        return False
```

All the best, give your honest attempt

SQ8. Write the output of the following function when we invoke it with countdown(4):

```
def countdown(n):  
    if n <= 0:  
        print ('Exam is over!')  
    else:  
        print (n)  
        countdown(n-1)
```

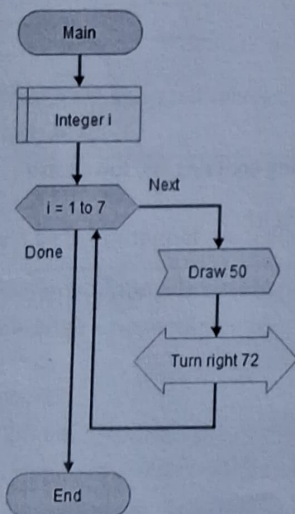
4  
3  
2  
Exam is over

SQ 9. Please write the sequence generated by the function when invoked with sequence(5).

```
def sequence(n):  
    while n != 1:  
        print (n)  
        if n%2 == 0: # n is even  
            n = n/2  
        else: # n is odd  
            n = n*3+1
```

5  
16  
8  
4  
2

SQ10. What shape is drawn by this Turtle graphics flowchart upon execution?



Pentagon