

Name \_\_\_\_\_; Roll no. \_\_\_\_\_; Section A; Group \_\_\_\_\_.

CSE 101 Introduction to Programming, **Section A MID-SEM Exam**, Sep 27, 2023

**Instructions:** **Part 1** There are 6 questions of 5 marks each, **Part 2:** There are 5 problems, each of 10 marks, for a total 80 marks. The total time allowed is 90 min.

- Where necessary, assume version of Python to be version 3.x.
- **Action will be taken against you if you are found to use unfair means.**
- You may use a calculator and a blank sheet to do rough work, **but NOT laptops, phones, notes, books, etc.**

**Part 1: There are 8 questions of 5 marks each**

1. Here is a definition of a function, `print_line(n1, n2)`. It is supposed to print a line such as those given below.

```
.  
--- *** ---
```

Or

```
- ***** -
```

```
def print_line(n1, n2):  
    print('-' * n1, '*' * n2, '-' * n1)
```

How should the function be called so that it prints the following pyramid:

```
---- * ----  
--- *** ---  
-- ***** --  
- ***** -
```

**Give your answer here:**

```
print_line(4, 1):  
print_line(3, 3):  
print_line(2, 5):  
print_line(1, 7):
```

2. What is the output of this Python program once it executes completely?

```
def recurse(n, s):  
    if n == 0:  
        print()  
    else:  
        print(n-1, s+n)  
        recurse(n-1, s+n)  
recurse(3, 0)
```

**Give your answer here:**

```
2 3  
1 5  
0 6
```

Name \_\_\_\_\_; Roll no. \_\_\_\_\_; Section A; Group \_\_\_\_\_.

3. You are given a Python program, where function **sqrt (K)** will return an approximate value of square root of **K**. What is the output once the main program is executed with input BIG = N in general, and when BIG = 57 in particular? **Write your answer here:**

This program prints the largest prime no. that is  $\leq N$ , where BIG = N.

In particular when BIG = 57, the program prints the largest prime no. 53 that is  $\leq 57$ .

```
def sqrt(K):  
    ...  
    return(root)  
def IS_Prime(N):  
    if N > 3:  
        max = int(sqrt(N))+1  
        composite = False  
        for k in range(2, max+1):  
            if N%k == 0:  
                composite = True  
                return(False)  
        if composite == False:  
            return(True)  
    else:  
        return(True)  
# The main program  
BIG = int(input('BIG: '))  
for M in range(BIG, 2, -1):  
    if IS_Prime(M):  
        print(BIG, M)  
        break
```

4. Below we have defined a function ABC(S1, S2) where S1 and S2 are strings. Then what will be returned when the function is executed with S1 and S2 given below:

```
def ABC(S1, S2):  
    if len(S1) == 0 or len(S2) == 0:  
        return(S1 + S2)  
    if S1[0] <= S2[0]:  
        return(S1[0] + ABC(S1[1:len(s1)], S2))  
    else:  
        return(S2[0] + ABC(S2[1:len(s2)], S1))
```

S1	S2	Value returned by the function
'art'	'grt'	'agrtrt'
'yellow'	'blow'	'blowyellow'
'hill'	'acer'	'acehillr'
"	'bdfh'	'bdfh'

Name \_\_\_\_\_; Roll no. \_\_\_\_\_; Section A; Group \_\_\_\_\_.

5. Take a look at the Python program given below. Fill the table below with variables and functions that are accessible locally, or those that are global to a given function.

```
def f():
    print(x)
def g():
    print(x)
    x = 1
x = 3
f()
x = 3
g()
```

	Local variables/objects	Global variables/objects
Main	x, f(), g()	-
f()	-	x, f(), g()
g()	x	f(), g()

6. Here is a piece of code in Python, dealing with strings. Which of the lines in the code are erroneous (viz. in error)?

```
x = "IP A" # line no. 1
y = "EASY CSE101-SecA" # line no. 2
print(x[0:2] + y[12:-1]) # line no. 3
print(x[-1] = "A") # line no. 4
print(y(len(y)) == "C") # line no. 5
```

**Give your answer here:**

Line 4, equality should be ==, and not =  
Line 5, y(len(y)) is unassigned

Name \_\_\_\_\_; Roll no. \_\_\_\_\_; Section A; Group \_\_\_\_\_.

**Part 2: There are 5 questions of 10 marks each**

7. Here is a definition of a function **ToH** to solve “Tower of Hanoi” puzzle, together with a function call with **ToH(3, src = 'A', dest = 'B', aux = 'C')**. What exactly is the output of the program.

```
def ToH(n, src, dest, aux):
    if n==1:
        print("Move disk 1 from peg ", src, "to peg ", dest)
        return
    ToH(n-1, src, aux, dest)
    print("Move disk", n, "from peg ",src, "to peg ", dest)
    ToH(n-1, aux, dest, src)
#
ToH(3, src = 'A', dest = 'B', aux = 'C')
```

**Give your answer here:**

```
Move disk 1 from peg A to peg B
Move disk 2 from peg A to peg C
Move disk 1 from peg B to peg C
Move disk 3 from peg A to peg B
Move disk 1 from peg C to peg A
Move disk 2 from peg C to peg B
Move disk 1 from peg A to peg B
```

Name \_\_\_\_\_; Roll no. \_\_\_\_\_; Section A; Group \_\_\_\_\_.

8. Write a function **factorial(n)** to calculate  $n!$ , where  $n > 0$ . Instead of using recursion, you have to use a for statement to implement this. You are also required to add statements in the main program to compute  $1!$ ,  $4!$ , and  $5!$

An outline of the overall program (**main** and function **factorial(n)**) is given below.

```
# Python program to compute n! and test it for 1!, 4! and 5!
def factorial(n):
# replace this line with one or more lines of your code to compute & return n!
# test statements
# replace this comment with one or more lines to compute and print 1!, 4! and 5!
```

**Answer:**

```
# Python program to compute n! and test it for 1!, 4! and 5!
def factorial(n):
    if n == 1:
        return(1)
    else:
        factorial = 1
        for k in range(2, n+1):
            factorial = factorial * k
        return(factorial)
# test statements
print(factorial(1))
print(factorial(4))
print(factorial(5))
```

Name \_\_\_\_\_; Roll no. \_\_\_\_\_; Section A; Group \_\_\_\_\_.

9. By definition: the definition of Fibonacci numbers,  $F(n)$ , is given below. Now write a function sub-program,  $F(n)$ , to compute the Fibonacci number,  $F(n)$  for  $n \geq 0$ . **BUT YOU MAY NOT USE RECURSION TO COMPUTE.**

$$\begin{aligned} F(n) &= 1, \text{ if } n = 0 \text{ or } 1, \\ &= F(n-1) + F(n-2), \text{ } n \geq 2 \end{aligned}$$

Correct answer:

```
# Python function to compute F(n), n >= 0
def Fibo(n):
    if n == 0 or n == 1:
        return(1)
    else:
        FNminus1 = 1
        FNminus2 = 1
        for k in range(2, n+1):
            FN = FNminus1 + FNminus2
            FNminus2 = FNminus1
            FNminus1 = FN
        return(FN)
```

It would be desirable for students to write one or two test statements, such as

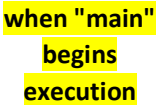
```
# test statements
print(Fibo(0))
print(Fibo(1))
print(Fibo(2))
print(Fibo(3))
print(Fibo(6))
```

A;

- "mxdexm"**. Use the format shown below to show the name of the frame ("main" or isPal") and values of parameters.

Name of actual parameter	Name of formal parameter
10	10
20	20
30	30
40	40
50	50
60	60
70	70
80	80
90	90
100	100

```
# main program
```



Name \_\_\_\_\_; Roll no. \_\_\_\_\_; Section A; Group \_\_\_\_\_.

11. Consider strings consisting of English letters only. Write a function, **reverse**, to reverse a given string, **S**, and return the reversed string, **T**. THEN write statements in the main program to reverse the input string, **S1**, and obtain string **T1**, and subsequently determine whether **S1** is a palindrome.

An outline of the program is given below. Replace the lines necessary to complete the program.

```
def reverse(S):  
    # replace this line to complete the body of the function of reverse  
    return(T)  
S1 = input('String S1 ')  
# replace this line in the main program & check if S1 is palindrome
```

```
def reverse(S):  
    T = ''  
    for c in S:  
        T = c + T  
    return(T)  
S1 = input('String S1 ')  
print('Is input string S1 a palindrome? ', S1 == reverse(S1))
```