Indian Association for the Cultivation of Science

(Deemed to be University under *de novo* Category)

Master's/Integrated Master's-PhD Program/Integrated Bachelor's-Master's Program/PhD Course

End-Semester Examination-Autumn 2022

Subject: Introduction to Computing with Python	***Subject Code(s): MCS1101B***
Full Marks: 50	***Time Allotted: 3 h***

Instructions *(please read carefully each point)*

- ★ Write as little as possible without missing out on any details
    - ○ Think carefully before answering
    - ○ There is no marks on being verbose
    - ○ Sometimes, adding an example makes things easier
- ★ There are **two sections** in the paper - (A) the **mandatory** section and (B) the **optional** section
- ★ You can answer *any combination of questions* from the ***mandatory Section A*** as long as the total marks attempted is *less than or equal to 50*, i.e., any question attempted beyond 50 marks will not be considered while marking
- ★ Additionally, you can attempt the ***optional Section B*** which contains *10 marks* in total; **this may be considered at the time of final grading but not for the End-sem exam itself**
- ★ If you are making any valid assumption(s) while writing an answer, do remember to mention that information clearly and concisely
- ★ 3 bonus marks will be awarded if there is *no minor mistakes* as well as *major ones, and questions worth a total of 50 marks are attempted*
- ★ Consider all questions in the **section A are for C language** and ***assume*** *the size of* ***int*** *and* ***float*** *as* ***4 bytes, char*** *as* ***1 byte, double*** *as* ***8 bytes, pointer types*** *as* ***8 bytes*** for this exam; also note the characters are evaluated using their **ASCII** values **A-Z** are valued **65-90** and **a-z** are valued **97-122** respectively
- ★ Questions marked with [Python] are all-or-nothing questions. For the rest, there will be part-marking.

Mandatory Section A

Q1.1. **Mark 2**

Write a function that takes a string as input and prints it.

Q1.2. **Mark 2**

Illustrate using a minimal example the concept of recursion.

Q1.3. **Mark 2**

```
int arr[2][2][2];
printf("%d %d %d %d", sizeof (arr), sizeof (arr[1]), sizeof (arr[1][0]), sizeof (arr[1][0][1][1]) );
```

Write down the output of the printf statement above.

Q1.4. **Mark 2**

```
struct new_type { int a; float b[3]; char name[10];}
struct new type n1;
printf ("%d %d", sizeof(n1), sizeof(name));
```

Write down the output of the printf statement above.

Q1.5. **Mark 2**

Create your own structure for storing points in a 4-dimensional space.

Q1.6. **Mark 2**

```
int fun(int* arr) {    printf ("in fun: %d\n", sizeof (arr));   }
int main()
{ int arr[10];   printf ("in main: %d\n", sizeof (arr));  fun (arr);  }
```

Write the output of code above.

Q1.7. **Mark 2**

Give code to return the absolute value of an integer. e.g. both -5 and 5 become 5.

Q1.8. **Mark 2**

```
double arr[4];
printf("%p", arr); ⇒ gives the output 0x1024
```

Calculate and write down the address of all the elements of the array arr.

Q1.9. **Mark 2**

Write a preprocessor(e.g. #define, #if, etc.) directive for getting the average of two values.

Q1.10. **Mark 2**

Write a simple code for opening and closing a file named “abc.txt” in write mode. *Just write the variable declaration(s) and the function call(s), no need to write #include, main, etc.*

Q1.11. **Mark 2**

Given a 2D coordinate position of a point, determine which quadrant the point is in. e.g. (-2,-2) is in the third quadrant, (2,3) is in the first quadrant.

Q1.12. **Mark 2**

Given an alphabet as input, check whether it is a vowel or a consonant.

Q2.1. **Mark 3**

```
void fun(int x, int y)
{
        if (x>y)
                fun(y+1, x-1);
        else if (x<y)
                fun(y-1, x+1);

        printf("%d %d\n", x, y);
}
```

Write the output when *fun(10,2)* is called.

Q2.2. **Mark 3**

Write a program/function to find the number of trailing zeros in a given factorial.

Example:
Input: 4 ⇒ Output: 0
Input: 10 ⇒ Output: 2
Input: 100 ⇒ Output: 24

Q2.3. **Mark 3**

```
float calc_avg (/*(A)formal parameter(s) for passing array*/);
int main ()
{
        float arr[] = {10.2, 10, 12, 11, 2, 3, 4, 9, 1, 2, 11.4, 2, 5, 19};
        int size = /*(B)write code for calculating number of elements*/;
        float avg = calc_avg(/*(C)pass the variable(s)*/);
}
```

Complete the above prototype and the corresponding function call for passing the array to the function. Just complete the A, B and C marked above. **Note**: You don't have to define the function, just assume it is already done.

Q2.4. **Mark 3**

Write a program/function to find the sum of the series 1!/(N-1) + 2!/(N-2) + 3!/(N-3) + 4!/(N-4) + …upto N terms. Take N as input form user.

Q2.5. **Mark 3**

You have two arrays of integers, each of size 5. Write code to create another array of size 10 containing all the elements in an alternating fashion.

Example:
Array 1: 10, 12, 14, 16, 18
Array 2: 9, 11, 13, 15, 17
Resultant array: 10, 9, 12, 11, 14, 13, 16, 18, 17

Q2.6. **Mark 3**

Show how to allocate memory using Dynamic memory allocation by allocating memory for an integer array of size N. N is read as input from the user.

Q3.1. **Mark 5**

Write a program/function that takes a month number (1-12) for the year 2023 as input and prints the dates for the Saturdays and Sundays in the month.

Example:
input 2 ⇒ Output: 4 5 11 12 18 19 25 26
input 9 ⇒ Output: 2 3 9 10 16 17 23 24 30

Hint: You need to identify the start day for the month, and you can/may use a 2D array

Q3.2. **Mark 5**

Write a program/function that takes a string as input and prints the upper case version of the string. Do not use library functions.

Example:
Input: souMAdiP ⇒ Output: SOUMADIP
Input: it’s a nice Day ⇒ Output: IT’S A NICE DAY
Input: 10 days ⇒ Output: 10 DAYS

Hint: loop, check for null characters, etc.

Q3.3. **Mark 5**

Write a C program to divide two integers (dividend and divisor) **without using** *multiplication*(*), *division*(/) and *modulo division*(%) operator.

Example:
input: 10 5 ⇒ Output: 2
input: 101 6 ⇒ Output: 5

Q3.4. **Mark 5**

```
typedef struct complex
{
        float real;
        float imaginary
}Q;
```

Write a function that takes the two complex numbers (you can use the above structure) and prints the multiplied value in x + yi format (check examples below, ignore 0s and treat 1i as i).

Hint: (x+yi)*(a+bi) = (ax-by) + (ay+bx)i
Example:
Input: 2 3 3 4 ⇒ Output: –6 + 17i
Input: 2 –3 3 4 ⇒ Output: 18 – i
Input: 1 2 1–2 ⇒ Output: 5
Input: 1 2 –2 –1 ⇒ Output: -5i

Optional Section B

Q4.1. **Mark 1**

[Python] Give an example of how to print a variable in python.

Q4.2. **Mark 1**

[Python] Give an example of how to do integer division in python. (e.g. 5÷2 = 2)

Q4.3. **Mark 1**

[Python] Give an example of how to write a list of integers in python.

Q4.4. **Mark 1**

[Python] Give an example of how to assign a value in a dictionary.

Q4.5. **Mark 1**

[Python] Give an example of how to access list elements using negative indexes.

Q4.6. **Mark 1**

[Python] A = [10, 12, 14, 16, 18, 20, 22, 24, 26, 28] ⇒ what is A[1:8]?

Q4.7. **Mark 1**

[Python] A = [10, 12, 14, 16, 18, 20, 22, 24, 26, 28] ⇒ what is A[1:7:2]?

Q4.8. **Mark 1**

[Python] A = [10, 12, 14, 16, 18, 20, 22, 24, 26, 28] ⇒ how do you reverse it?

Q4.9. **Mark 1**

[Python] Assume x = “Hi” and y=”There” ⇒ what will be x+y?

Q4.10. **Mark 1**

[Python] How do you calculate the length of the string “sly fox”?