INDIAN ASSOCIATION FOR THE CULTIVATION OF SCIENCE

SCHOOL OF MATHEMATICAL & COMPUTATIONAL SCIENCES

PG - I (SMCS), Autumn Semester 2022-23

Instructor: **Debarshi Kumar Sanyal** Assignment Number: **1**

Course: **COM 4111: Object-Oriented Programming with C++** Date: August 22, 2022

August 22, 2022

INSTRUCTIONS

(1) Create a folder with the name `Assignment_1_$name$` (where `$name$` denotes name of the student). In that folder create sub-folders for each problem. (example: `prob_0`, `prob_1`, `prob_2`, etc.) Write your codes for a particular problem in the respective folder (i.e., `prob_1` should contain two C programming files `swap.c` and `swap_2.c`)

(2) Give meaningful comments to explain the method used in your program. Programs without valid comments will be evaluated to zero marks.

Problem 0

(i) Run the given code and explain the outputs.

```
#include<stdio.h>
int main()
{
    int a = 1, b = 2, c = 3, d = 4, res = 0;
```

```
    res = a+b-c+d;
    printf("res = %d \n",res);
    res = a*b/c;
    printf("res = %d \n",res);
    res = 1+a*b%c;
    printf("res = %d \n",res);
    res = a+d%b-c;
    printf("res = %d \n",res);
    res = b = d+c/b-a;
    printf("res = %d \n",res);
    return 0;
}
```

(ii) Run the given code and explain the outputs.

```
#include<stdio.h>

int main()
{
int ans, val=4;
val = val+1;
printf("ans=%d val=%d\n",ans,val);
val++;
++val;
printf("ans=%d val=%d\n",ans,val);
ans = 2*val++;
printf("ans=%d val=%d\n",ans,val);
val--;
--val;
printf("ans=%d val=%d\n",ans,val);
ans=--val*2;
printf("ans=%d val=%d\n",ans,val);
ans = val--/3;
printf("ans=%d val=%d\n",ans,val);
return 0;
}
```

Problem 1 Write a C program which reads three **real valued** coefficients a, b and c of a quadratic equation $ax^2+bx+c = 0$ and compute and print the roots. (All cases must be covered.) [`quadratic.c`]

Problem 2 Write a C program to perform addition and multiplication of two large integers of size N and M where N and M are user inputs. Note that the integers are too large to be stored in built-in types. [`large_numbers.c`]

Problem 3 Write a C program which reads two integers n and r and find the value of $\binom{n}{r}$ by defining a function named as `factorial` which returns the factorial of an integer. [`nCr.c`]

Problem 4 Write a C program to compute the sum of the following series for a given value of x $x - \frac{x^3}{3!} + \frac{x^5}{5!} + ...$, upto an approximation of 10^{-5} [`sum_A.c`]

Problem 5 Write a C program to calculate the integral of a function in a given range. [`integral.c`]

Problem 6 Write a C program that can compute the *multiplication* of two matrices, and report if not possible. [`matrix_mult.c`]

Problem 7

(i) Write a C function `mySubStr` which takes two character strings `s1, s2` and checks whether the second string is a substring of the latter. If so, it should return the position (the index value) of the occurrence of `s2` in `s1`,

(ii) Write a C program which reads two strings, prints them and uses the function `mySubStr` to detect and report the first occurrence of the second string in the first one. [`string_search.c`]

Problem 8 Use `typedef` to define a data type `COMPLEX` to represent complex numbers. Write C functions:

(a) `COMPLEX add(COMPLEX, COMPLEX)` to *add* two complex numbers,

(b) `COMPLEX mult(COMPLEX, COMPLEX)` to *multiply* two complex numbers.

Write a suitable main function to test the above functions. [`complex.c`]

Problem 9

(i) Run the given code and explain the outputs. [`basic_pointers.c`]

```
#include <stdio.h>
int main()
{
    int x, y;
    int *ptr;

    x = 10 ;
    ptr = &x ;
    y = *ptr ;
    printf ("%d is stored in location %p \n", x, &x) ;
    printf ("%d is stored in location %p \n", *&x, &x) ;
```

```
    printf ("%d is stored in location %p \n", *ptr, ptr) ;
    printf ("%d is stored in location %p \n", y, &*ptr) ;
    printf ("%p is stored in location %p \n", ptr, &ptr) ;
    printf ("%d is stored in location %p \n", y, &y) ;

    *ptr = 25;
    printf ("\nNow x = %d \n", x);
    return 0;
}
```

(ii) Write a C program that can swap two integers using pointers. [swap_pointers.c]

(iii) Write a C program to read any two integer values m and n from the keyboard. Then allocate the memory to store a two-dimensional matrix, say, `DynamicMatrix[m][n]`. Initialize the matrix so defined. Then print the matrix in the matrix form and also display the sum of the rows and the sum of the columns of the matrix. [matrix_pointers.c]

Problem 10 Write a C program (using recursion) to print all permutations of an N-digit number input by the user. Example: [large_numbers.c]

```
Input: 132
Output: 123, 132, 213, 231, 312, 321
```