Dept. of CSE, Bennett University

ECSE379L – Programming Using C++

Lab Assignment – 4

In this lab, you will write a C++ program to solve the following problem.

(1) Given two positive integer arrays A[] and B[].

  a. First task is to find all possible pair wise sum and store in an array C[].

  b. Second is to find the bitwise xor of all pairs of elements present in array C[] and print the maximum among them (MAXPair).

Sample Input: A[] = {1,2}, B[] = {3,4}
Sample Output: (a) C[] = {4, 5, 5, 6}, (b) MAXPair = 3

(2) Write a program to find the area of Triangle, Rectangle and Sphere using function overloading. Call the corresponding function to find area of each shape and return the area to the main function and print the area of the shape in main function only. (You need to ask the user to input the choice i.e. which shape's area the user wants to calculate, Use if – elseif – else to call the respective functions.

(3) Any given number (N) can be written in the form like $P1^{E1}$ * $P2^{E2}$ * $P3^{E3}$ *... *Pn^{En} where Pi is the prime factor of N and Ei is it's exponent. Here, Pi, Ei >= 1. The task is to take input any number and display it in the specified format as above.

Sample Input: 36
Sample Output: 2^2*3^2

Sample Input: 91
Sample Output: 7^1*13^1

(4) We can store a 2D 10 x 10 square matrix in a 1D array of size 100 by sequentially writing each row one after another. Now, it is also given that the (i, j)-th element is equal to the (j, i)-th element of the 2D matrix for all i and j. This means that the given matrix is a symmetrical matrix around its diagonal.

So, as an efficient programmer, we need not store all the 100 elements in the 1D array. So, there are two possibilities. Either we can store the lower triangle or the upper triangle.

Let us assume that we will store only the lower triangle of the 2D Matrix. Then, in the 1D array we will store in the following manner. The first element of the first row of the 2D Matrix, then first two elements of the second row of the 2D Matrix, then first three elements of the third row of the 2D Matrix, and so on.

Let us assume that we will store only the upper triangle of the 2D Matrix. Then, in the 1D array we will store in the following manner. The n elements of the first row of the 2D Matrix, then last (n-1) elements of the second row of the 2D Matrix, then last (n-2) elements of the third row of the 2D Matrix and so on.

Now, you write a program to perform the above operation and based on the user's choice do the first solution or the second solution. And subsequently, take the 2D matrix as input. And finally display the corresponding 1D matrix.

Sample Input: Choice of solution = 1

11	2	3	4	5	6
2	22	7	8	10	11
3	7	33	9	12	13
4	8	9	44	14	15
5	10	12	14	55	16
6	11	13	15	16	66

Sample Output: {11, 2, 22, 3, 7, 33, 4, 8, 9, 44, 5, 10, 12, 14, 55, 6, 11, 13, 15, 16, 66}

Submission Instructions:

- Submit your assignment files from LMS within 7 days from the day of your lab slot. Save all the files as per the format RollNo_Lab#_QuestionNo.docx (Example: E18CSE362_Lab4_Q2.docx). Make a single .zip file and upload at a time.
- In the LMS please submit in your respective batch's submission portal. Submission in other batch's submission portal will not be checked.
- Write your Name and Roll No in each page of the design itself. Without this you will score zero for that particular question.
- Provide label/comments in the appropriate place.
- Late submission will lead to penalty.
- Any form of plagiarism/copying from peer or internet sources will lead penalty.