**Final Assignment**
**Full marks: 30**

**There are 15 problems and each weighs 2 marks. Good Luck.**

(1) Given a decimal number ( <= 10^9) , convert it into a Hexa-Decimal Number.

| **Sample Input** | **Sample Output** |
|---|---|
| 4 | 4 |
| 10 | A |
| 12 | C |
| 30 | 1E |
| 255 | FF |

(2) Given two sets of integers of distinct integers of length n and m, print their union.

**Time complexity of your algorithm must be O((n+m)logn) or better.**

| **Sample Input** | **Sample Output** |
|---|---|
| 4<br>3 2 1 7<br>7<br>1 2 3 4 5 10 11 | 1 2 3 4 5 7 10 11 |

Explanation:
n = 4, Set, A = [3, 2, 1, 7]
m = 7, Set, B = [1, 2, 3, 4, 5, 10, 11]
A U B = [1, 2, 3, 4, 5, 7, 10, 11]

(3) Given A and B, find A%B without using the modulus operator (%).

| **Sample Input** | **Sample Output** |
|---|---|
| 10 12 | 10 |
| 100 12 | 4 |
| 1009 12 | 1 |
| 10 5 | 0 |

(4) You are given an array of integers of size n. Each element of the array is between 1 to n. **You have to sort the array in non-descending order O(n) time complexity**.

| **Sample Input** | **Sample Output** |
|---|---|
| 4<br>3 3 2 4 | 2 3 3 4 |

Note that you can not just use the c++ sort function, as the time complexity for that would be O(nlogn).

(5) You are given a binary string of size N of form : “000…...1111”. Find the index (0-based) of the first 1 in the string in **O(logN) time complexity**.
It is guaranteed that there is at least one 1 in the string.

| **Sample Input** | **Sample Output** |
|---|---|
| 000011 | 4 |

(6) You’ll be given an integer array of size n, for each index you have to answer the product of the whole array except this number. You can’t use the division operation.

**You need to solve the problem in O(n) time complexity.**

| Sample Input | Output |
|---|---|
| 4<br>2 -3 1 4 | -12 8 -24 -6 |

Explanation:
1st number of output is : (-3)*1*4 = -12
2nd number of output is : 2*1*4 = 8
3rd number of output is : 2*(-3)*4 = -24
4th number of output is : 2*(-3)*1 = -6

| | |
|---|---|
| 5<br>2 0 1 -2 9 | 0 -36 0 0 0 |

(7) Given an array $a(a_1a_2….a_n)$ of size n and an integer x, you’ve to answer how many subarrays have sum equal to x.
$1<=n<=10^6$ , $1<=a_i<=10^9$ , $1<=x<=10^9$

**Your complexity should be better than or equal to O(nlogn)** .

| Sample Input | Sample Output |
|---|---|
| 5 3<br>4 -2 5 -3 3 | 3 |

Explanation: {-2,5} , {-2,5,-3,3} , {3} are all the subarrays having sum 3.

(8) There is a board of **NxM** size. Each cell of the board is either
   (a) ‘x’ : the cell is blocked
   (b) ‘o’ : the cell is free

(c) ‘s’ : starting cell
(d) ‘e’ : exit cell

Starting and ending cells are also considered free.

There is a knight (a chess piece) on the board occupying the start cell. Can you find the minimum number of moves required to reach the exit cell ?

| **Sample Input** | **Sample Output** |
|---|---|
| 4 4<br>oxox<br>sxoe<br>ooxo<br>oooo | 3 |

**Explanation:**
The first line contains N, M.
Here N = 4, M = 4

The optimal movements of the knight are shown in the following,

| | | | |
|---|---|---|---|
| oxox | ox**o**x | oxox | oxox |
| **s**xoe → | sxoe → | sxoe → | sxo**e** |
| ooxo | ooxo | o**o**xo | ooxo |
| oooo | oooo | oooo | oooo |

(9) Given x, find ln(x) upto 5 decimal places. **You must not use log/log10 functions but you can use exp function.**
Here, ln(x) represents the natural logarithm of x.

| Sample Input | Sample Output |
|---|---|
| 1 | 0.00000 |
| 5 | 1.60944 |
| 10 | 2.30259 |
| 100 | 4.60517 |

(10) Print all binary strings of length n such that no two adjacent characters are ‘1’.

| Sample Input | Samput Output |
|---|---|
| 4 | 0000<br>0001<br>0010<br>0100 |

0101
1000
1001
1010

(11) Given an array $a(a_1a_2….a_n)$ of size n and an integer k, generate all the subsequences of the array such that the product of that subsequence is divisible by k.
1<=n,k<=20
**No constraints on complexity.**

| Sample Input | Sample Output |
|---|---|
| 4 6<br>9 4 7 10 | 9,4<br>9,10<br>9,4,10<br>9,4,7<br>9,7,10<br>9,4,7,10 |

(12) Given an n*m grid containing integers. Rows are numbered from 1 to n and columns are numbered from 1 to m. Find the path with minimum sum from (1,1) to (n,m). You can either move right or down at each step. 1<= (n,m) <= 1000.

**Your complexity should be O(n*m) .**

| Sample Input | Sample Output |
|---|---|
| 4 5<br>2 -3 8 8 5<br>7 -2 5 8 3<br>2 1 3 0 4<br>3 4 2 8 4 | 9 |

Explanation: 2 -> -3 -> -2 -> 1 -> 3 -> 0 -> 4 -> 4

(13) Given n coins $c(c_1c_2…..c_n)$ and a sum s. Find out the minimum number of coins required to make s and print those coins. You can use each coin multiple times.
1 <= n <= 100 , 1<=s <= $10^6$
**You've to solve it in O(n*s) .**

| Sample Input | Sample Output |
|---|---|
| 5 23<br>7 5 4 9 15 | 3<br>5 9 9 |

(14) Given two integer arrays $a(a_1a_2...a_n)$ and $b(b_1b_2....b_n)$ of size n, find out the longest increasing common subsequence among them.
(1<=n<=500) and all the array elements are between 1 to n.
**Your complexity should be equal or better than $O(n^3)$.**

| Sample Input | Sample Output |
|---|---|
| 5<br>2 3 1 4 5<br>4 2 3 2 5 | 3 |

Explanation: (2,3,5) is the longest increasing common subsequence among the arrays.

(15) You've n jobs to do, i'th job starts at time $s_i$, ends at time $e_i$ and it will bring you profit $p_i$. As you are not good at multitasking, you can do only one task at a time. If you start a job, you can't switch to another task until it ends but once it ends, you can switch to another job instantly. Find out the maximum amount of profit you can make.
$1<=n<=10^5$, $1 <= s_i < e_i <= 2*n$, $1 <= p_i <= 10^9$
**You've to solve it in O(n) .**

| Sample Input | Sample Output |
|---|---|
| 5<br>2 4 7<br>3 8 8<br>1 2 2<br>5 7 5<br>9 10 6 | 20 |

Explanation: You can do all the jobs except 2nd one and earn 7+2+5+6 = 20.