# MCA-1 Programming Lab Set-5: Miscellaneous C/C++ Coding Assignments

✓ 1. Write a program to multiply 2 integers a & b (take input from user).

Constraint:  $1 \le a, b \le 10^{1000}$ .

# Sample Runtime Environment:

Enter a

2049874329569346590314032475981765481654981644659031654916541403247598176 5481654984659031654916541403247598176548165490498743204987432049874328465 9031654916541403247598176548165498465903165491654140324759817654816549840 4987432049874326590316654916541403247598176548165498164 (user input)

#### Enter b:

5491654140324759817654816549816454916541403247598176548165498164549165414
0324759817654816549816446590316549165414032475981765481654984659031654916
5414032475981765481654904987432049874320498743284659031654916541403247598
1765481654984659031654916541403247598176548165498404987432049874326590316
5491654140324759817654816549846590316549165414032475981765481654980498743
2 (user input)

#### Multiplication Result:

 $11257200849124943433279170670345590296394826431637982736894156783614047460\\ 6069555061521996602639841369507753377321390258486884847575965766127193320\\ 9955053202525647820400414708801983276020521062487451802209996536703195310\\ 2897905672581374722214270241537451528342328644195340629616583748184183010\\ 9553987916601419343086276192729083176107434065623277401910202389884620483\\ 00710540110084113161819115299586866213373807327629104014067572104564200428\\ 19390546124148789927935018864897890367228383203452775119066448885949800268\\ 32538335818666222106066306040997994899237987666011526181379195399211680401\\ 8370032720799941115386857377456084006684714039074848$ 

\*You can follow your own output formatting but it must be correct and interpretable.

2. Write a program to factorize the polynomial,  $ax^3 + bx^2 + cx + d$ , where a,b,c & d are integers and to be taken as user input. The constraints are:  $-20 \le a, b, c, d \le 20$  and  $a \ne 0$ . Input consists only one line of 4 space separated values a, b, c & d respectively.

## Sample Runtime Environment:

Enter the coefficients of the polynomial: 9 0 0 -9 (user input)
The Polynomial is: 9x^3 - 9 (you can write it as 9x^3 + 0x^2 + 0x^1 - 9x^0)

No. of factors: 3 Factor#1: 9

Factor#2: x-1 (you can write it as 1x^1 - 1x^0)

Factor#3:  $x^2 + x + 1$  (you can write it as  $1x^2 + 1x^1 + 1x^0$ )

\*You can follow your own output formatting but it must be correct and interpretable.

✓ 3. A traveler has to climb to the top of the hill with the least possible cost. The hill climbing process and the corresponding cost are described below.

The hill is represented by a 2-D binary matrix of order m×n ( $1 \le m, n \le 50$ ) in which 1 means the traveler can go to that spot (cell), 0 means the traveler cannot go to the spot. The traveler can directly go to its neighbor spot in the right or left direction with cost 2, upper direction with cost 4, lower direction with cost 1. He cannot move diagonally. Starting from the bottom-left corner, the traveler has to reach the maximum reachable height with the least possible cost.

Print the maximum reachable height of the hill and the least possible cost to reach there.

# Sample Runtime Environment:

Enter the dimension of the hill matrix: 9 8 (i.e. m=9 & n=8)

Enter the hill matrix:

00001010

01010101

10101111

11111010

10101111

10100110

01111010

00011101

1 1 1 1 1 1 1 1 (user input up to this)

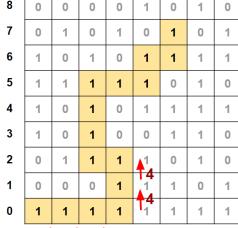
Maximum reachable height: 7

Minimum cost to reach the maximum reachable height: 42

### Explanation:

<u>Height</u>

8	0	0	(
7	0	1	(



Cost: 2 2

The figure in the left explains everything. The cost of the first 5 moves of the traveller is written. Calculate the other costs according to the moves & sum it up to get the total cost.

Note: There can be other ways to get to the maximum reachable height. Print the least cost to reach there.

4. Find the number of binary strings of length n, such that '0' must be a neighbor of '1' and '1' must be a neighbor of '0'. Take n ( $2 \le n \le 1000$ ) as user input. Print the answer in 'modulo  $10^9 + 7$ ' format.

For example, for string length n=5, '01101' will be counted but '10001' won't be counted as the middle '0' is NOT satisfying the condition.

5. Print factored polynomial form of f(k) where,  $f(k) = 1^k + 2^k + 3^k + ... + n^k$ , the value of the integer k ( $1 \le k \le 20$ ) is to be taken as user input.

Test Case	Input <b>k</b>	Output <b>f(k)</b>
1	1	n*(n+1)/2
2	3	n*n*(n+1)*(n+1)/4
3	6	n*(n+1)*(2n+1)*(3n^4 + 6n^3 - 3n + 1)/42

# Sample Runtime Environment:

Enter k: 2

f(k) is: n(n+1)(2n+1)/6

**Hint:** Initially you have to find the simple polynomial form of f(k). Then you need to factorize it as you did in the 2nd assignment (But this factorization will be a generic version of what you did in the 2nd assignment for degree 3 only).

<sup>\*</sup>You can follow your own output formatting but it must be correct and interpretable.