Fundamentals of Computing

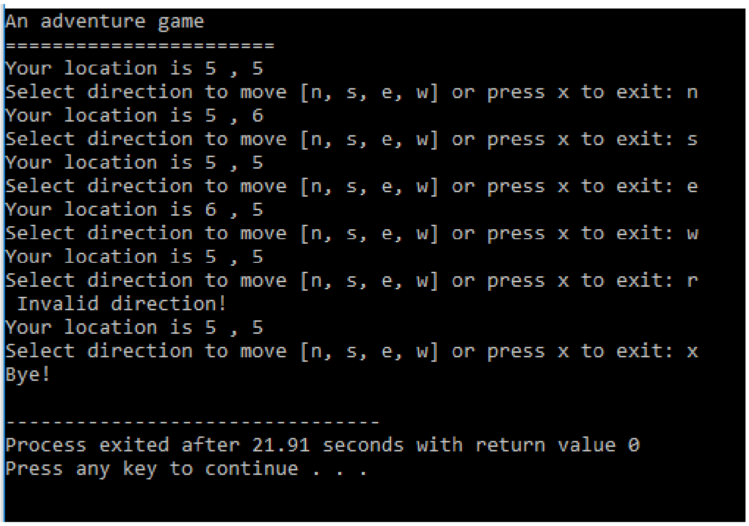
Laboratory Exercise 5

1) Multi-User Dungeon

During the 70s, an online computer game named MUD or Multi-User Dungeon was created and played by gamers. It is a multiplayer real-time virtual world text-based game. The traditional MUDs implement a role-playing adventure game set in a fantasy world populated by fictional races and monsters, where players choose their class in order to gain specific skills or powers. You are probably too young to remember or even play adventure games on early character-mode DOS systems. But, let us restore the concept here.

Suppose you are tasked to create a program of an adventure game in a DOS system where user can move the character around an imaginary landscape. The game starts in a deserted grassland with coordinates 5,5. The user can go one step north, south, east or west. Your program should be able to keep track of where the user is and report its position or coordinates. The program should also track for invalid input. Additionally, it will keep on asking for the direction unless the user press x to exit from the program.

Sample Run:



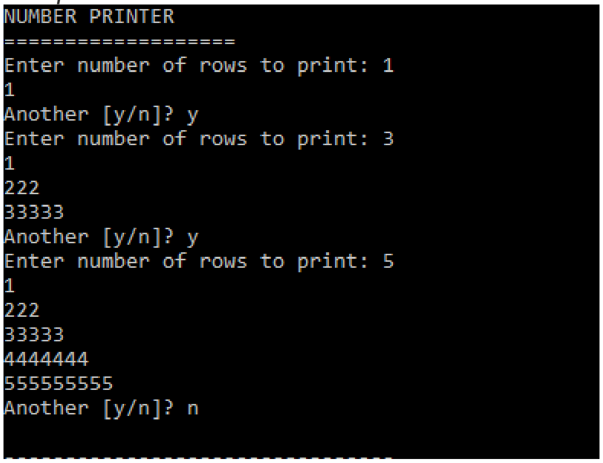
```
An adventure game
=======================
Your location is 5 , 5
Select direction to move [n, s, e, w] or press x to exit: n
Your location is 5 , 6
Select direction to move [n, s, e, w] or press x to exit: s
Your location is 5 , 5
Select direction to move [n, s, e, w] or press x to exit: e
Your location is 6 , 5
Select direction to move [n, s, e, w] or press x to exit: w
Your location is 5 , 5
Select direction to move [n, s, e, w] or press x to exit: r
 Invalid direction!
Your location is 5 , 5
Select direction to move [n, s, e, w] or press x to exit: x
Bye!

--------------------------------
Process exited after 21.91 seconds with return value 0
Press any key to continue . . .
```

2) Number Printer

The program should be able to print the numbers in triangle shape based on the number of rows entered by the user. Your program should allow users to print the numbers again.

Sample Run:



```
NUMBER PRINTER
===================
Enter number of rows to print: 1
1
Another [y/n]? y
Enter number of rows to print: 3
1
222
33333
Another [y/n]? y
Enter number of rows to print: 5
1
222
33333
4444444
555555555
Another [y/n]? n

--------------------------------
```

3) Reverse

Write a C++ program that will read up to ten letters into an array and print the letters back to the screen in reverse order.

Sample Input:

```
abcd.
```

Sample Output:

```
dcba
```

*Use period as a sentinel value to mark the end of the input. Call the array letterBox.

4) Even From End

Given a list of integers, find and display all even numbers from the end of the list.

INPUT FORMAT:

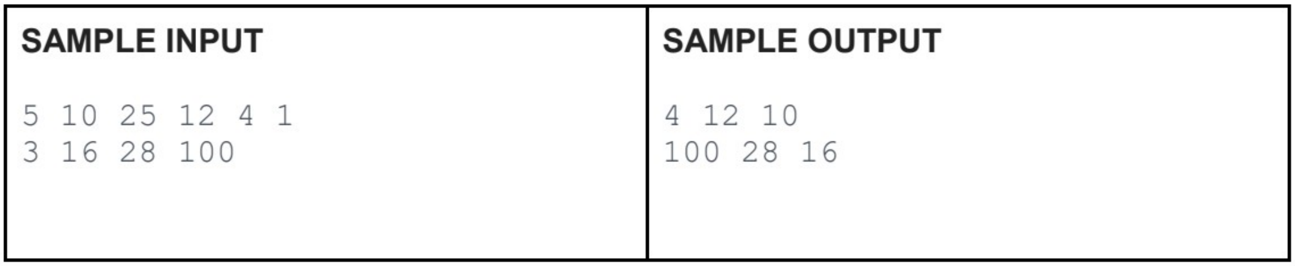
Each line of input begins with an integer indicating the number of integer that follow which comprises a list

OUTPUT FORMAT:

All even numbers from the end of the list, each separated by a single space. Separate output for each test case with the newline character.

CONSTRAINTS:

$$2 <= N <= 100$$
$$-1000 <= n <= 1000$$



SAMPLE INPUT	SAMPLE OUTPUT
5 10 25 12 4 1 3 16 28 100	4 12 10 100 28 16