CS19101 PDS laboratory
Lab Test-1

[Odd PC]

Write programs for problems 1,2 and 3 in three different files named LT1_1_<machine number>_<Roll no.>.c, LT1_2_<machine number>_<Roll no.>.c and LT1_3_<machine number>_<Roll no.>.c respectively (without the '<' and '>'). Put these three files into a compressed directory named LT1_<machine number>_<Roll no.>.zip and submit it.

Example: If your roll number is 19DEP99999 and your machine number is 99, then the names of your files should be LT1_1_99_19DEP99999.c, LT1_2_99_19DEP99999.c and LT1_3_99_19DEP99999.c.

---

1. Take an integer as input through keyboard, and print a pattern on the screen as shown below.

```
Enter integer:5
*               *
* *           * *
* * *       * * *
* * * *   * * * *
* * * * * * * * *
* * * *   * * * *
* * *       * * *
* *           * *
*               *
```

```
Enter integer:4
*           *
* *       * *
* * *   * * *
* * * * * * *
* * *   * * *
* *       * *
*           *
```

[10 marks]

2. In this problem do not use array. The user first enters an integer a through keyboard. Then the user will enter co-efficients of a univariate polynomial one by one, starting with the constant term (see sample input/output). After the user enters each co-efficient, ask the user to enter 1 if she wants to enter more coefficient, and 0 otherwise. After the user enters as many co-efficients as she likes, print the evaluation of the resulting polynomial on the integer a on the screen.

   Sample input/output:

   ```
   Enter a: 5
   Enter coefficient of 1: -2
   Enter more co-efficient? 1
   Enter co-efficient of x to the power 1: 2
   Enter more co-efficient? 1
   Enter co-efficient of x to the power 2: -2
   Enter more co-efficient? 0
   -42
   ```

3. Write a function with the following prototype:
   int split(int A[]);
   The above function takes in an integer array as input. It returns an index i such that $A[0] + \ldots + A[i] = A[i + 1] + \ldots + A[9]$ if such an index i ($0 \leqslant i \leqslant 9$) exists, and returns -1 otherwise. In main(), declare an integer array of size 10. Fill the array by taking inputs through the keyboard. Then pass the array to split(). Finally in main() print what split() returns.

   Sample input/output:

   ```
   Enter number 1: -2
   Enter number 2: 3
   Enter number 3: 0
   Enter number 4: 5
   Enter number 5: 8
   Enter number 6: 9
   ```

```
Enter number 7: -4
Enter number 8: 6
Enter number 9: 9
Enter number 10: 12
5
```

```
Enter number 1: 2
Enter number 2: 2
Enter number 3: 2
Enter number 4: 2
Enter number 5: 2
Enter number 6: 2
Enter number 7: -4
Enter number 8: 6
Enter number 9: 9
Enter number 10: 12
-1
```