IRC_JAVA_CODING CONTEST 2

Test Summary

- No. of Sections: 1
- No. of Questions: 5
- Total Duration: 60 min

Section 1 - Coding

Section Summary

- No. of Questions: 5
- Duration: 60 min

Additional Instructions:

None

Q1. Decode the logic and print the pattern that corresponds to the given input.

Input

5

Output

```
1     *
2    **
3   *  *
4  *    *
5 *      *
6  *    *
7   *  *
8    **
9     *
```

Input Format

The input will contain a single integer

Output Format

Print the pattern mentioned in the problem statement.

Sample Input

```
5
```

Sample Output

```
    *
   * *
  *   *
 *     *
*       *
```

Sample Input

```
4
```

Sample Output

```
   *
  * *
 *   *
*     *
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. Decode the logic and print the pattern that corresponds to the given input.

Input

5

Output

```
1 *****
2   *
3   *
4   *
5 *****
```

Input Format

The input will contain a single integer

Output Format

Print the pattern mentioned in the problem statement.

Sample Input

```
5
```

Sample Output

```
*****
    *
   *
  *
 *
```

Sample Input

```
4
```

Sample Output

```
****
   *
  *
 *
****
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. Decode the logic and print the pattern that corresponds to the given input.

Input
5

Output
5 5 5 5 5
4 5 5 5 5
3 4 5 5 5
2 3 4 5 5
1 2 3 4 5

Input Format

The input will contain a single integer n denoting the number of rows.

Output Format

Print the pattern mentioned in the problem statement.

Sample Input

```
5
```

Sample Output

```
55555
45555
34555
23455
```

Sample Input

```
4
```

Sample Output

```
4444
3444
2344
1234
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. Decode the logic and print the pattern that corresponds to the given input.

Input
4

Output
1 * 2 * 3 * 4
9 * 10 * 11 * 12
13 * 14 * 15 * 16
5 * 6 * 7 * 8

Input Format

The input will contain a single integer n denoting the number of rows.

Output Format

Print the pattern mentioned in the problem statement.

Sample Input

```
4
```

Sample Output

```
1 * 2 * 3 * 4
9 * 10 * 11 * 12
13 * 14 * 15 * 16
5 * 6 * 7 * 8
```

Sample Input

```
5
```

Sample Output

```
1 * 2 * 3 * 4 * 5
11 * 12 * 13 * 14 * 15
21 * 22 * 23 * 24 * 25
16 * 17 * 18 * 19 * 20
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. Write a program to arrange the given sorted array in an order such that the first largest element in the first position, first smallest element in the second position,second largest element in the third position and so on.

Input Format

Input to get an integer N in first line,second line contains N number of elements separated by single space.

Output Format

Display the output as shown in the sample output.

Sample Input

```
8
2 3 4 5 6 7 8 9
```

Sample Output

```
9 2 8 3 7 4 6 5
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - Coding

Q1

Test Case

Input

```
6
```

Output

```
     *
    * *
   *   *
  *     *
```

Weightage - 25

Input

```
3
```

Output

```
  *
 * *
*   *
 * *
```

Weightage - 25

Input

```
7
```

Output

```
      *
     * *
    *   *
   *     *
```

Weightage - 25

Input

```
10
```

Output

```
        *
       * *
      *   *
     *     *
```

Weightage - 25

Sample Input

```
5
```

Sample Output

```
    *
   * *
  *   *
 *     *
```

Sample Input

```
4
```

Sample Output

```
   *
  * *
 *   *
*     *
```

Solution

```
import java.util.Scanner;
class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int rows = sc.nextInt();
        for (int i=1; i<= rows ; i++)
```

```
{
    for (int j = rows; j > i ; j--)
    {
        System.out.print(" ");
    }
    System.out.print("*");
    for (int k = 1; k < 2*(i -1) ;k++)
    {
        System.out.print(" ");
    }
    if( i==1)
    {
        System.out.println("");
    }
    else
    {
        System.out.println("*");
    }

}
for (int i=rows-1; i>= 1 ; i--)
{
            for (int j = rows; j > i ; j--)
            {
                System.out.print(" ");
            }
            System.out.print("*");
            for (int k = 1; k < 2*(i -1) ;k++)
            {
                System.out.print(" ");
            }
            if( i==1)
                System.out.println("");
            else
                System.out.println("*");
        }
    }
}
```

Q2

Test Case

Input	Output

6	******
	     *
	    *
	   *

Weightage - 25

Input	Output
2	** **

Weightage - 25

Input	Output

7	*******
	      *
	     *
	    *

Weightage - 25

Input

```
10
```

Output

```
**********
         *
        *
       *
```

Weightage - 25

Sample Input

```
5
```

Sample Output

```
*****
    *
   *
  *
```

Sample Input

```
4
```

Sample Output

```
****
   *
  *
****
```

Solution

```
#include<stdio.h>
int main(){
    int n;
    scanf("%d", &n);
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            if((i + j) == (n + 1) || i == n || i == 1){
                printf("*");
            }
            else{
                printf(" ");
            }
        }
        printf("\n");
    }
}
```

Q3

Test Case

Input

```
3
```

Output

```
333
233
123
```

Weightage - 25

Input

```
7
```

Output

```
7777777
6777777
5677777
4567777
```

Weightage - 25

Input

```
9
```

Output

```
999999999
899999999
789999999
678999999
```

Weightage - 25

Input

```
1
```

Output

```
1
```

Weightage - 25

Sample Input

```
5
```

Sample Output

```
55555
45555
34555
23455
```

Sample Input

```
4
```

Sample Output

```
4444
3444
2344
1234
```

Solution

```
import java.util.Scanner;
class A
{
    void d()
    {
Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        int i,j,k;

      for(i=a;i>=1;i--)
        {

        for(j=i;j<=a;j++)
        {
            System.out.print(j+"");

        }

        for(k=1;k<=i-1;k++)
          {
              System.out.print(a+"");

        }
        System.out.println("");

        }
    }
}
class Main
{
    public static void main(String args[])
```

```
    {

     A c=new A();
     c.d();
    }
}
```

Q4

Test Case

Input

```
3
```

Output

```
1 * 2 * 3
7 * 8 * 9
4 * 5 * 6
```

Weightage - 25

Input

```
6
```

Output

```
1 * 2 * 3 * 4 * 5 * 6
13 * 14 * 15 * 16 * 17 * 18
25 * 26 * 27 * 28 * 29 * 30
31 * 32 * 33 * 34 * 35 * 36
```

Weightage - 25

Input

```
7
```

Output

```
1 * 2 * 3 * 4 * 5 * 6 * 7
15 * 16 * 17 * 18 * 19 * 20 * 21
29 * 30 * 31 * 32 * 33 * 34 * 35
43 * 44 * 45 * 46 * 47 * 48 * 49
```

Weightage - 25

Input

```
2
```

Output

```
1 * 2
3 * 4
```

Weightage - 25

Sample Input

```
4
```

Sample Output

```
1 * 2 * 3 * 4
9 * 10 * 11 * 12
13 * 14 * 15 * 16
5 * 6 * 7 * 8
```

Sample Input

```
5
```

Sample Output

```
1 * 2 * 3 * 4 * 5
11 * 12 * 13 * 14 * 15
21 * 22 * 23 * 24 * 25
16 * 17 * 18 * 19 * 20
```

Solution

```
#include<bits/stdc++.h>
using namespace std;

int main()
```

```
{
    int n;

    cin>>n;
    int p=n;

    for(int i=1;i<=n;i+=2)
    {
        int k=(i-1)*n+1;
        for(int j=0;j<n-1;j++)
        {
            printf("%d * ",k);
            k++;
        }
        printf("%d \n",k);
    }
    if(n%2!=0)
    {
        p=n-1;
    }
    for(int i=p;i>0;i-=2)
    {
        int k=(i-1)*n+1;
        for(int j=0;j<n-1;j++)
        {
            printf("%d * ",k);
            k++;
        }
        printf("%d \n",k);
    }

    return 0;
}
```

Q5

Test Case

Input

```
6
3 4 5 6 7 8
```

Output

```
8 3 7 4 6 5
```

Weightage - 20

Input

```
10
12 13 14 15 16 17 34 56 67 89
```

Output

```
89 12 67 13 56 14 34 15 17 16
```

Weightage - 20

Input

```
2
3 3
```

Output

```
3 3
```

Weightage - 30

Input

```
4
1  2  3  4
```

Output

```
4  1  3  2
```

Weightage - 10

Input

```
6
23  24  25  26  27  28
```

Output

```
28  23  27  24  26  25
```

Weightage - 10

Input

```
5
8  12  23  24  35
```

Output

```
35  8  24  12  23
```

Weightage - 10

Sample Input

```
8
2  3  4  5  6  7  8  9
```

Sample Output

```
9  2  8  3  7  4  6  5
```

Solution

```
 import java.util.Arrays;
import java.util.Scanner;
class Main
{
    Main(int[] arr, int n)
    {
        int temp[] = new int[n];
    int small=0, large=n-1,i;
    boolean flag = true;
    for (i=0; i<n; i++)
    {
        if (flag)
            temp[i] = arr[large--];
        else
            temp[i] = arr[small++];

        flag = !flag;
    }
  arr = temp.clone();
  for(i=0;i<n;i++){
  System.out.print(arr[i]+" ");
  }
}
public static void main(String[] args)
{
    int i,n;
    Scanner in =new Scanner(System.in);
    n=in.nextInt();
    int arr[] = new int[n];
```

```
        for(i=0;i<n;i++){
        arr[i]=in.nextInt();
        }
        Main obj=new Main(arr,arr.length);

    }
}
```