

Name:-Sahil Kumar

Mobile:-9162231102

Email: sahilkumar.118969@marwadiuiversity.ac.in

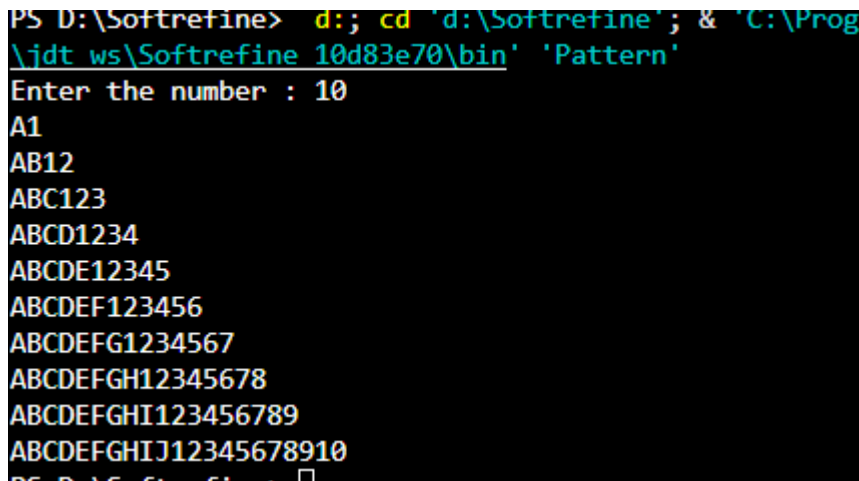
Collage: Marwadi University

Degree: B.Tech(CE)

Start:3.11pm

Question: Write a program to generate the following pattern

Sol.

A screenshot of a command prompt window with a black background and green text. The prompt shows the command to run a Java program named 'Pattern' from the 'bin' directory of a specific user. The program prompts the user to 'Enter the number : 10'. It then prints a pattern of alphanumeric strings. Each line consists of a sequence of letters followed by a sequence of numbers, with the total length of the string increasing by one in each subsequent line. The letters start at 'A' and the numbers start at '1'. The pattern is as follows:
A1
AB12
ABC123
ABCD1234
ABCDE12345
ABCDEF123456
ABCDEFG1234567
ABCDEFGH12345678
ABCDEFGHI123456789
ABCDEFGHIJ12345678910

Code snippet:

```
import java.util.*;
public class Pattern{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number : ");
        int number = sc.nextInt();
        for(int i=0; i<number; i++){
            int Alphabet = 65;
            for(int j=0; j<=i; j++){
                System.out.print((char) Alphabet++);
            }
            for(int j=0; j<=i; j++){
                System.out.print(j+1);
            }
            System.out.println();
        }
    }
}
```

```
}  
sc.close();  
}  
}  
End:3.31pm
```

Start:3.43pm

Question: Given A square matrix, write a program to rotate it in place by 90 degree.

Sol.

```
PS D:\Softrefine> d:; cd 'd:\Softrefine'; & 'C:\5992\redhat.java\jdt_ws\Softrefine_10d83e70\bin'
Enter the Matrix
1 2 3
4 5 6
7 8 9

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

Code snippet:

```
import java.util.*;
public class Matrix {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = 3;
        System.out.println("Enter the Matrix ");
        int[][] Matrix = new int[n][n];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                Matrix[i][j] = sc.nextInt();
            }
        }

        for (int j = n - 1; j >= 0; j--) {
            for (int i = 0; i < n; i++) {
                System.out.print(Matrix[i][j]+" ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

End: 3.58pm

Start:4.12pm

Question: Find longest Balanced String

Sol.

```
PS D:\Softrefine> d:; cd 'd:\Softrefine'; & 'C:\Program Files\Java\jdk-5992\redhat.java\jdt_ws\Softrefine_10d83e70\bin' 'LongestBalancedString'
Enter the String :
<<?
2
0
2
PS D:\Softrefine> d:; cd 'd:\Softrefine'; & 'C:\Program Files\Java\jdk-5992\redhat.java\jdt_ws\Softrefine_10d83e70\bin' 'LongestBalancedString'
Enter the String :
??????
0
0
6
PS D:\Softrefine> d:; cd 'd:\Softrefine'; & 'C:\Program Files\Java\jdk-5992\redhat.java\jdt_ws\Softrefine_10d83e70\bin' 'LongestBalancedString'
Enter the String :
??<<>>??
0
2
8
```

Code snippet:

```
import java.util.*;

public class LongestBalancedString {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the String : ");

        String s = sc.nextLine();

        int length = s.length();

        Stack<Character> stack = new Stack<>();

        int number = 0;
```

```
int count = 0;
for (int i = 0; i < length; i++) {
    if (s.charAt(i) == '?') {
        count++;
    }
}
for (int i = 0; i < length; i++) {
    if (s.charAt(i) == '<') {
        number++;
        stack.push(s.charAt(i));
    }
    if (stack.peek() == '<' && s.charAt(i) == '>') {
        stack.pop();
        number--;
    }
    if (stack.isEmpty() && s.charAt(i) == '?') {
        stack.push('<');
        number++;
        count--;
    }
    if (stack.peek() == '<' && s.charAt(i) == '?') {
        if (i < count) {
            stack.push('>');
            number++;
            count--;
        }
    }
}
```

```
    }  
    System.out.println(number);  
    sc.close();  
}  
}
```

Incomplete

Start:5:45pm

Question: Longest Common Subsequence

Sol.

```
PS D:\Softrefine> d:; cd 'd:\Softrefine'; & 'C:\Program Files\Java\jdk1.8.0_205\bin\java.exe' -cp 'C:\Program Files\Java\jdk1.8.0_205\redhat.java\jdt_ws\Softrefine_10d83e70\bin' 'LongestCommonSubsequence'
Enter the Text1 : abcde
Enter the Text2 : abc
3
PS D:\Softrefine> d:; cd 'd:\Softrefine'; & 'C:\Program Files\Java\jdk1.8.0_205\bin\java.exe' -cp 'C:\Program Files\Java\jdk1.8.0_205\redhat.java\jdt_ws\Softrefine_10d83e70\bin' 'LongestCommonSubsequence'
Enter the Text1 : abc
Enter the Text2 : abc
3
PS D:\Softrefine> d:; cd 'd:\Softrefine'; & 'C:\Program Files\Java\jdk1.8.0_205\bin\java.exe' -cp 'C:\Program Files\Java\jdk1.8.0_205\redhat.java\jdt_ws\Softrefine_10d83e70\bin' 'LongestCommonSubsequence'
Enter the Text1 : abc
Enter the Text2 : def
0
PS D:\Softrefine> d:; cd 'd:\Softrefine'; & 'C:\Program Files\Java\jdk1.8.0_205\bin\java.exe' -cp 'C:\Program Files\Java\jdk1.8.0_205\redhat.java\jdt_ws\Softrefine_10d83e70\bin' 'LongestCommonSubsequence'
Enter the Text1 : abcdcdcbcd
Enter the Text2 : abcdcd
```

Code snippet:

```
import java.util.*;

public class LongestCommonSubsequence {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Text1 : ");

        String text1 = sc.nextLine();

        int length1 = text1.length();

        System.out.print("Enter the Text2 : ");

        String text2 = sc.nextLine();
```



```
int length2 = text2.length();
int[][] dp = new int[length1 + 1][length2 + 1];
for (int i = 1; i <= length1; i++) {
    for (int j = 1; j <= length2; j++) {
        if (text1.charAt(i - 1) == text2.charAt(j - 1)) {
            dp[i][j] = dp[i - 1][j - 1] + 1;
        } else {
            dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]);
        }
    }
}
System.out.println(dp[length1][length2]);
}
```

End: 6:03pm