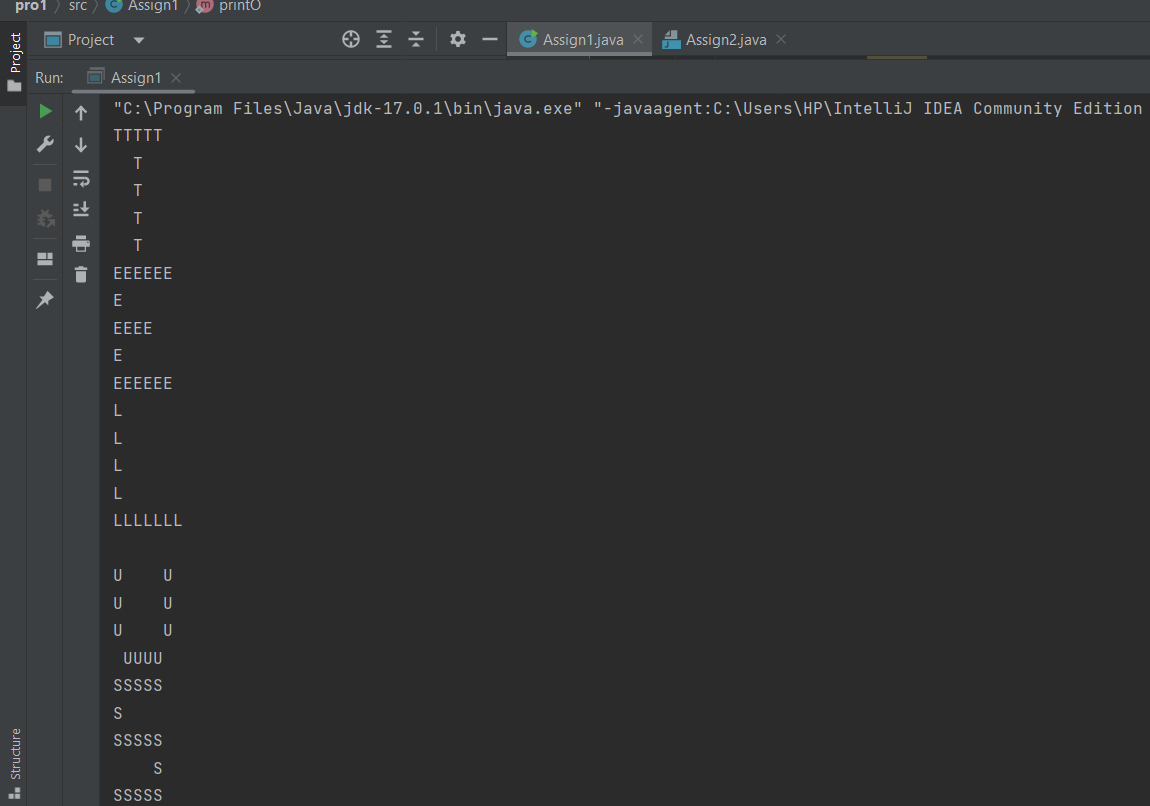
public class Assign1 {  
// Below height and width variable can be used  
// to create a user-defined sized alphabet's pattern  
  
 // Number of lines for the alphabet's pattern  
 static int *height* = 5;  
  
 // Number of character width in each line  
 static int *width* = (2 \* *height*) - 1;  
  
 // Function to find the absolute value  
// of a number D  
 static int abs(int d)  
 {  
 return d < 0 ? -1 \* d : d;  
 }  
 static void printT()  
 {  
 int i, j;  
 for (i = 0; i < *height*; i++)  
 {  
 for (j = 0; j < *height*; j++)  
 {  
 if (i == 0)  
 System.*out*.printf("T");  
 else if (j == *height* / 2)  
 System.*out*.printf("T");  
 else  
 System.*out*.printf(" ");  
 }  
 System.*out*.print("\n");  
 }  
 }  
 static void printE()  
 {  
 int i, j;  
 for (i = 0; i < *height*; i++)  
 {  
 System.*out*.printf("E");  
 for (j = 0; j < *height*; j++)  
 {  
 if ((i == 0 || i == *height* - 1)  
 || (i == *height* / 2  
 && j <= *height* / 2))  
 System.*out*.printf("E");  
 else  
 continue;  
 }  
 System.*out*.printf("\n");  
 }  
 }  
 static void printL()  
 {  
 int i, j;  
 for (i = 0; i < *height*; i++)  
 {  
 System.*out*.printf("L");  
 for (j = 0; j <= *height*; j++)  
 {  
 if (i == *height* - 1)  
 System.*out*.printf("L");  
 else  
 System.*out*.printf(" ");  
 }  
 System.*out*.printf("\n");  
 }  
 }  
 static void printU()  
 {  
 int i, j;  
 for (i = 0; i < *height*; i++)  
 {  
 if (i != 0 && i != *height* - 1)  
 System.*out*.printf("U");  
 else  
 System.*out*.printf(" ");  
 for (j = 0; j < *height*; j++)  
 {  
 if (((i == *height* - 1)  
 && j >= 0  
 && j < *height* - 1))  
 System.*out*.printf("U");  
 else if (j == *height* - 1 && i != 0  
 && i != *height* - 1)  
 System.*out*.printf("U");  
 else  
 System.*out*.printf(" ");  
 }  
 System.*out*.printf("\n");  
 }  
 }  
 static void printS()  
 {  
 int i, j;  
 for (i = 0; i < *height*; i++)  
 {  
 for (j = 0; j < *height*; j++)  
 {  
 if ((i == 0 || i == *height* / 2  
 || i == *height* - 1))  
 System.*out*.printf("S");  
 else if (i < *height* / 2  
 && j == 0)  
 System.*out*.printf("S");  
 else if (i > *height* / 2  
 && j == *height* - 1)  
 System.*out*.printf("S");  
 else  
 System.*out*.printf(" ");  
 }  
 System.*out*.printf("\n");  
 }  
 }  
 static void printK()  
 {  
 int i, j, half = *height* / 2, dummy = half;  
 for (i = 0; i < *height*; i++)  
 {  
 System.*out*.printf("K");  
 for (j = 0; j <= half; j++)  
 {  
 if (j == *abs*(dummy))  
 System.*out*.printf("K");  
 else  
 System.*out*.printf(" ");  
 }  
 System.*out*.printf("\n");  
 dummy--;  
 }  
 }  
 static void printO()  
 {  
 int i, j, space = (*height* / 3);  
 int width = *height* / 2 + *height* / 5 + space + space;  
 for (i = 0; i < *height*; i++)  
 {  
 for (j = 0; j <= width; j++)  
 {  
 if (j == width - *abs*(space)  
 || j == *abs*(space))  
 System.*out*.printf("O");  
 else if ((i == 0  
 || i == *height* - 1)  
 && j > *abs*(space)  
 && j < width - *abs*(space))  
 System.*out*.printf("O");  
 else  
 System.*out*.printf(" ");  
 }  
 if (space != 0  
 && i < *height* / 2)  
 {  
 space--;  
 }  
 else if (i >= (*height* / 2 + *height* / 5))  
 space--;  
 System.*out*.printf("\n");  
 }  
 }  
  
 static void printPattern(char character) {  
 switch (character) {  
 case 'T':  
 *printT*();  
 break;  
 case 'E':  
 *printE*();  
 break;  
 case 'L':  
 *printL*();  
 break;  
 case 'U':  
 *printU*();  
 break;  
 case 'S':  
 *printS*();  
 break;  
 case 'K':  
 *printK*();  
 break;  
 case 'O':  
 *printO*();  
 break;  
  
 }  
 }  
  
 private static void printF() {  
 }  
  
 public static void main(String[] args)  
 {  
 *printPattern*('T');  
 *printPattern*('E');  
 *printPattern*('L');  
 *printPattern*('U');  
 *printPattern*('S');  
 *printPattern*('K');  
 *printPattern*('O');  
  
 }  
 }

OUTPUT-

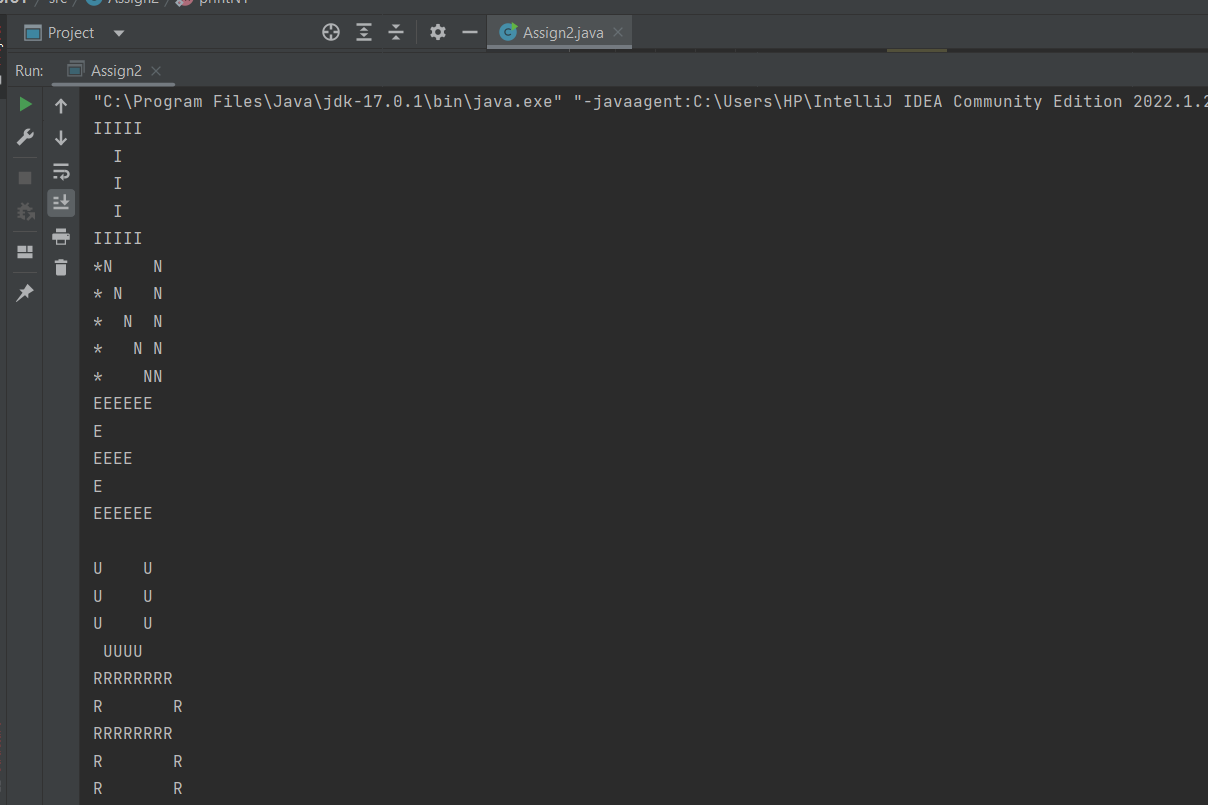




2.)

class Assign2  
{  
  
// Below height and width variable can be used  
// to create a user-defined sized alphabet's pattern  
  
 // Number of lines for the alphabet's pattern  
 static int *height* = 5;  
  
 // Number of character width in each line  
 static int *width* = (2 \* *height*) - 1;  
  
 // Function to find the absolute value  
// of a number D  
 static int abs(int d)  
 {  
 return d < 0 ? -1 \* d : d;  
 }  
 static void printI()  
 {  
 int i, j;  
 for (i = 0; i < *height*; i++)  
 {  
 for (j = 0; j < *height*; j++)  
 {  
 if (i == 0 || i == *height* - 1)  
 System.*out*.printf("I");  
 else if (j == *height* / 2)  
 System.*out*.printf("I");  
 else  
 System.*out*.printf(" ");  
 }  
 System.*out*.printf("\n");  
 }  
 }  
 static void printN()  
 {  
 int i, j, counter = 0;  
 for (i = 0; i < *height*; i++)  
 {  
 System.*out*.printf("\*");  
 for (j = 0; j <= *height*; j++)  
 {  
 if (j == *height*)  
 System.*out*.printf("N");  
 else if (j == counter)  
 System.*out*.printf("N");  
 else  
 System.*out*.printf(" ");  
 }  
 counter++;  
 System.*out*.printf("\n");  
 }  
 }  
 static void printE()  
 {  
 int i, j;  
 for (i = 0; i < *height*; i++)  
 {  
 System.*out*.printf("E");  
 for (j = 0; j < *height*; j++)  
 {  
 if ((i == 0 || i == *height* - 1)  
 || (i == *height* / 2  
 && j <= *height* / 2))  
 System.*out*.printf("E");  
 else  
 continue;  
 }  
 System.*out*.printf("\n");  
 }  
 }  
 static void printU()  
 {  
 int i, j;  
 for (i = 0; i < *height*; i++)  
 {  
 if (i != 0 && i != *height* - 1)  
 System.*out*.printf("U");  
 else  
 System.*out*.printf(" ");  
 for (j = 0; j < *height*; j++)  
 {  
 if (((i == *height* - 1)  
 && j >= 0  
 && j < *height* - 1))  
 System.*out*.printf("U");  
 else if (j == *height* - 1 && i != 0  
 && i != *height* - 1)  
 System.*out*.printf("U");  
 else  
 System.*out*.printf(" ");  
 }  
 System.*out*.printf("\n");  
 }  
 }  
 static void printR()  
 {  
 int i, j, half = (*height* / 2);  
 for (i = 0; i < *height*; i++)  
 {  
 System.*out*.printf("R");  
 for (j = 0; j < *width*; j++)  
 {  
 if ((i == 0 || i == half)  
 && j < (*width* - 2))  
 System.*out*.printf("R");  
 else if (j == (*width* - 2)  
 && !(i == 0 || i == half))  
 System.*out*.printf("R");  
 else  
 System.*out*.printf(" ");  
 }  
 System.*out*.printf("\n");  
 }  
 }  
 static void printO()  
 {  
 int i, j, space = (*height* / 3);  
 int width = *height* / 2 + *height* / 5 + space + space;  
 for (i = 0; i < *height*; i++)  
 {  
 for (j = 0; j <= width; j++)  
 {  
 if (j == width - *abs*(space)  
 || j == *abs*(space))  
 System.*out*.printf("O");  
 else if ((i == 0  
 || i == *height* - 1)  
 && j > *abs*(space)  
 && j < width - *abs*(space))  
 System.*out*.printf("O");  
 else  
 System.*out*.printf(" ");  
 }  
 if (space != 0  
 && i < *height* / 2)  
 {  
 space--;  
 }  
 else if (i >= (*height* / 2 + *height* / 5))  
 space--;  
 System.*out*.printf("\n");  
 }  
 }  
 static void printN1()  
 {  
 int i, j, counter = 0;  
 for (i = 0; i < *height*; i++)  
 {  
 System.*out*.printf("N");  
 for (j = 0; j <= *height*; j++)  
 {  
 if (j == *height*)  
 System.*out*.printf("N");  
 else if (j == counter)  
 System.*out*.printf("N");  
 else  
 System.*out*.printf(" ");  
 }  
 counter++;  
 System.*out*.printf("\n");  
 }  
 }  
 static void printPattern(char character) {  
 switch (character) {  
 case 'I':  
 *printI*();  
 break;  
 case 'N':  
 *printN*();  
 break;  
 case 'E':  
 *printE*();  
 break;  
 case 'U':  
 *printU*();  
 break;  
 case 'R':  
 *printR*();  
 break;  
 case 'O':  
 *printO*();  
 break;  
 }  
 }  
 private static void printF() {  
 }  
  
 public static void main(String[] args)  
 {  
 *printPattern*('I');  
 *printPattern*('N');  
 *printPattern*('E');  
 *printPattern*('U');  
 *printPattern*('R');  
 *printPattern*('O');  
 *printPattern*('N');  
  
 }  
}

OUTPUT-



DONE BY

K.R.PRANAV KUMAR