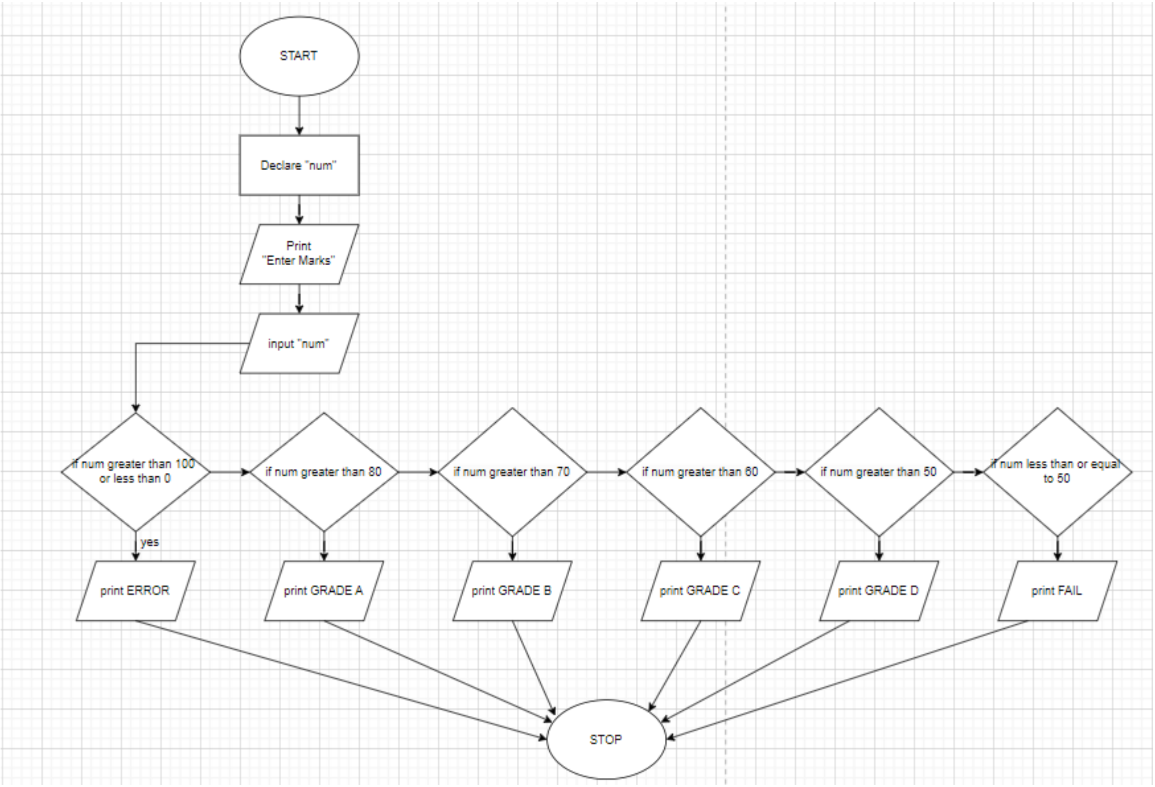
Name	Lekh Sanatan Nayak
UID no.	223800068
Experiment No.	2

AIM:	Apply various control structures to solve given problems
Program 1	
PROBLEM STATEMENT :	Find the letter grade of a student marks using if else.
ALGORITHM:	1. Start 2. Declare variable num. 3. Display “print marks” and store user input in num variable using scanf function. 4. Check if user input is within range, if it is not within range print “ERROR”. 5. If the user input is within range and greater than 80 print “GRADE A”. 6. If the user input is greater than 70 print “GRADE B”. 7. If the user input is greater than 60 print “GRADE C”. 8. If the user input is greater than 50 print “GRADE D”. 9. If the user input is less than or equal to 50 print “FAIL”. 10. Stop

FLOWCHART:

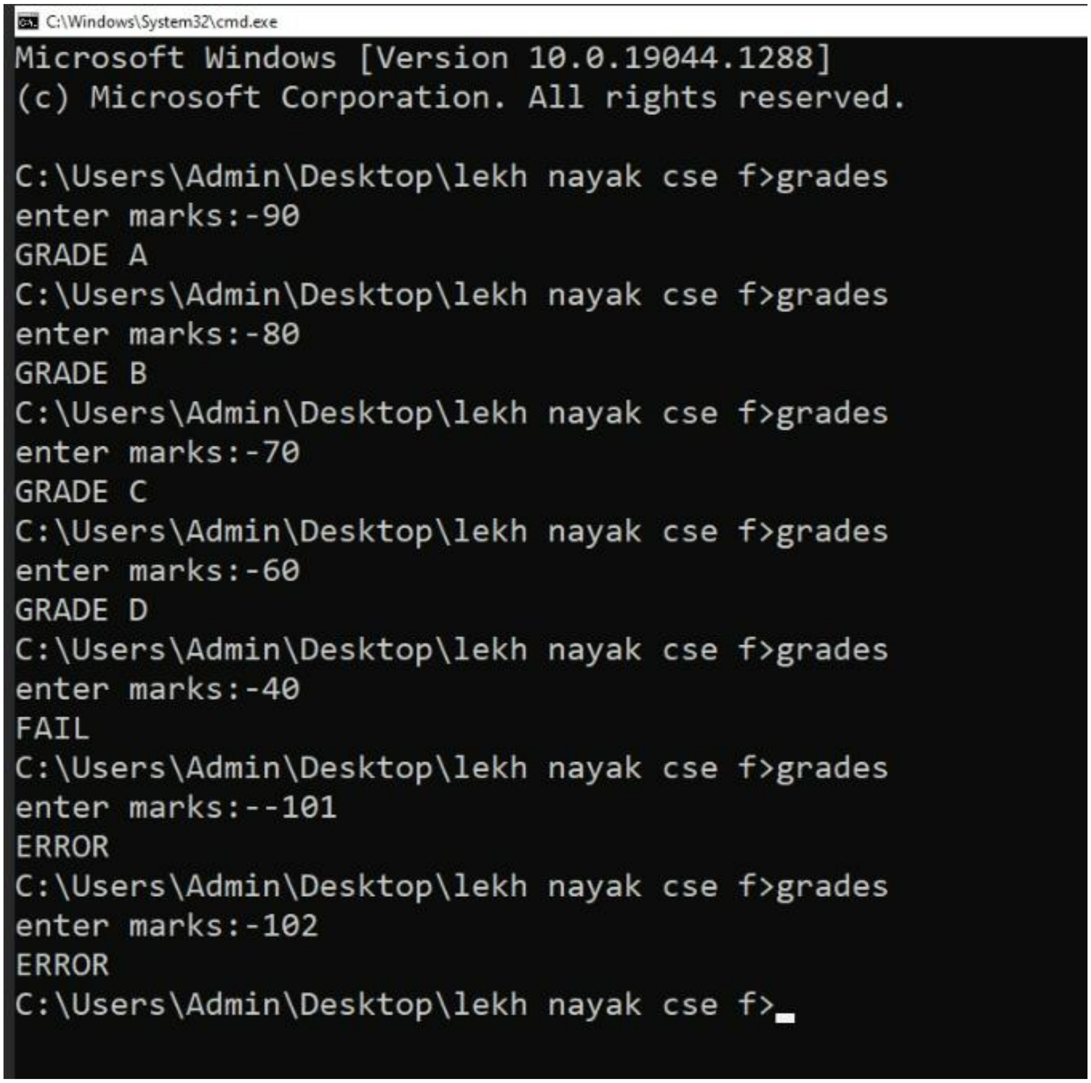


PROGRAM:

```
#include<stdio.h>
int main(){
int num;
printf("enter marks:-");
scanf("%d",&num);

if(num>100 || num<0){
  printf("ERROR");
  return 0;
}

if(num>80){
  printf("GRADE A");
}
else if(num>70){
  printf("GRADE B");
}
else if(num>60){
  printf("GRADE C");
}
else if(num>50){
  printf("GRADE D");
}
else if(num<=50){
  printf("FAIL");
}
return 0;
```

}

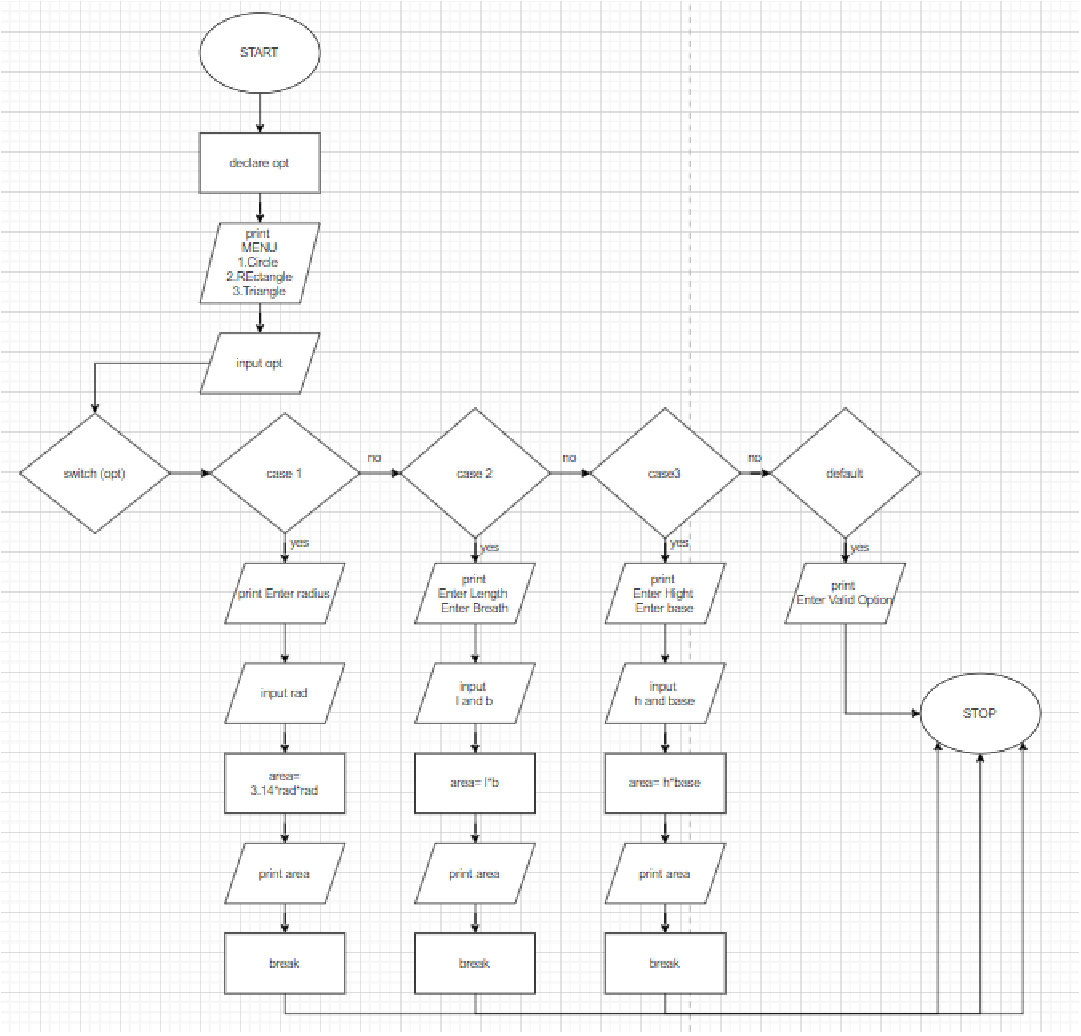
RESULT:



```
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\lekh nayak cse f>grades
enter marks:-90
GRADE A
C:\Users\Admin\Desktop\lekh nayak cse f>grades
enter marks:-80
GRADE B
C:\Users\Admin\Desktop\lekh nayak cse f>grades
enter marks:-70
GRADE C
C:\Users\Admin\Desktop\lekh nayak cse f>grades
enter marks:-60
GRADE D
C:\Users\Admin\Desktop\lekh nayak cse f>grades
enter marks:-40
FAIL
C:\Users\Admin\Desktop\lekh nayak cse f>grades
enter marks:--101
ERROR
C:\Users\Admin\Desktop\lekh nayak cse f>grades
enter marks:-102
ERROR
C:\Users\Admin\Desktop\lekh nayak cse f>
```

Program 2	
PROBLEM STATEMENT :	Write a program using switch case to find the area of different shapes.
ALGORITHM:	1. Start 2. Declare variable “opt”. 3. Display the menu of options to the user and prompt the user to input the number corresponding their choice using scanf function. 4. Read and store the user input in “opt” variable. 5. Use a switch statement to perform different actions based the value of user input. 6. If user input is 1 print Enter radius and take user input in variable rad and calculate area or circle with the formula 3.14*rad*rad, print the area and break out of the switch function 7. If user input is 2 print Enter length and breath and take user input and store them in variables l and b and calculate area off rectangle using the formula l*b, print the area and break out of the switch function 8. If user input is 3 print Enter height and base and take user input and store them in variables h and base and calculate the area of triangle using the formula 0.5*b*h, print the area and break out of the switch function 9. If user input is invalid print Enter valid input 10. End switch function 11. Stop

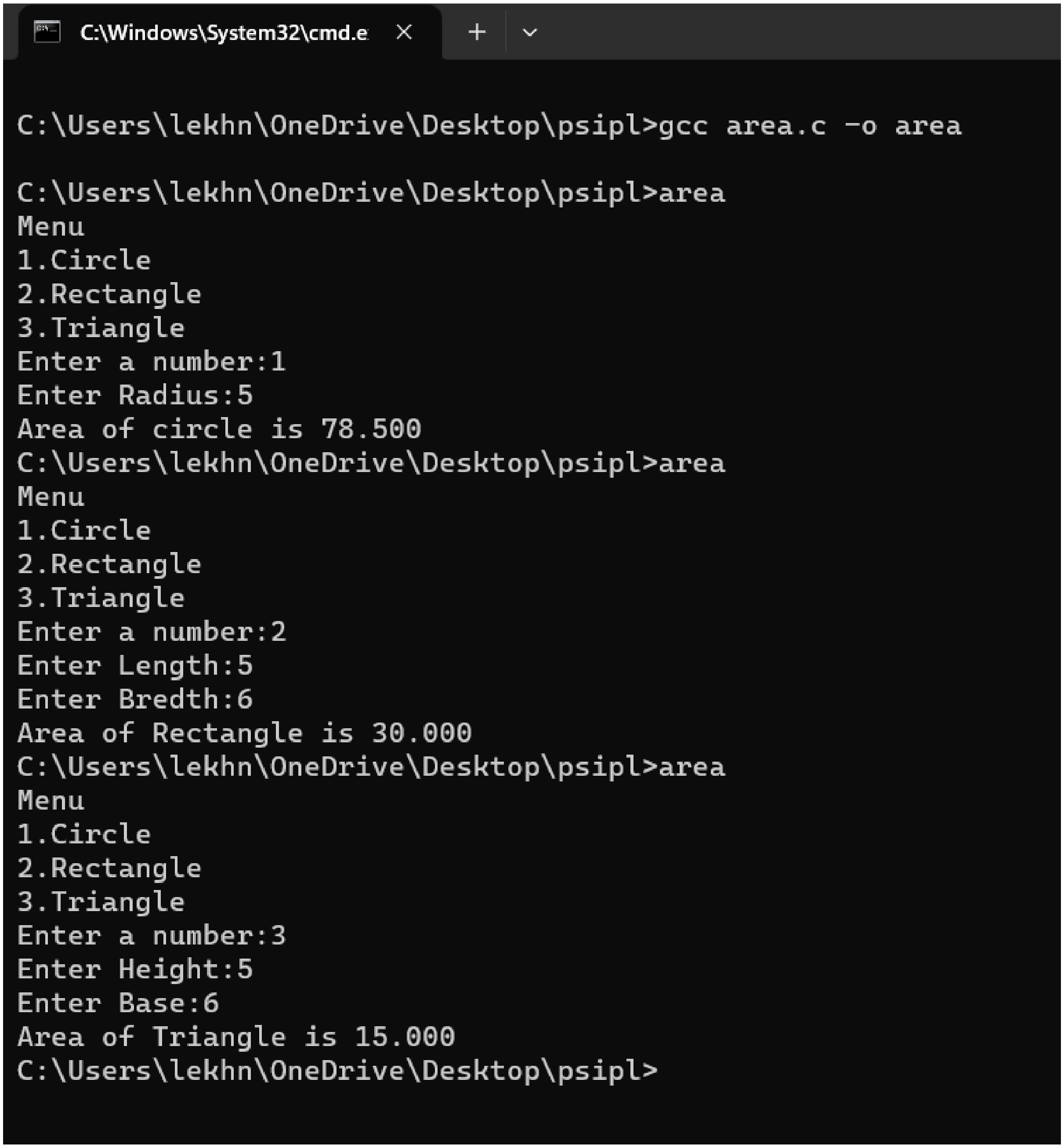
FLOWCHART:



PROGRAM:

```
#include <stdio.h>

int main()
{
    int opt;
    printf("Menu\n1.Circle\n2.Rectangle\n3.Triangle\nEnter a number:");
    scanf("%d",&opt);



    switch (opt)
    {
    case 1:{
        float rad;
        printf("Enter Radius:");
        scanf("%f", &rad);
        printf("Area of circle is %.3f", rad*rad*3.14);
        break;
}

    case 2:{
        float l,b;
        printf("Enter Length:");
        scanf("%f", &l);
        printf("Enter Bredth:");
        scanf("%f", &b);
        printf("Area of Rectangle is %.3f", l*b);
        break;}

    case 3:{
        float h,base;
        printf("Enter Height:");
        scanf("%f", &h);
        printf("Enter Base:");
        scanf("%f", &base);
        printf("Area of Triangle is %.3f", 0.5*h*base);
        break;}


    default:
        printf("Enter valid Option");
        break;
    }
    return 0;
}
```

RESULT:



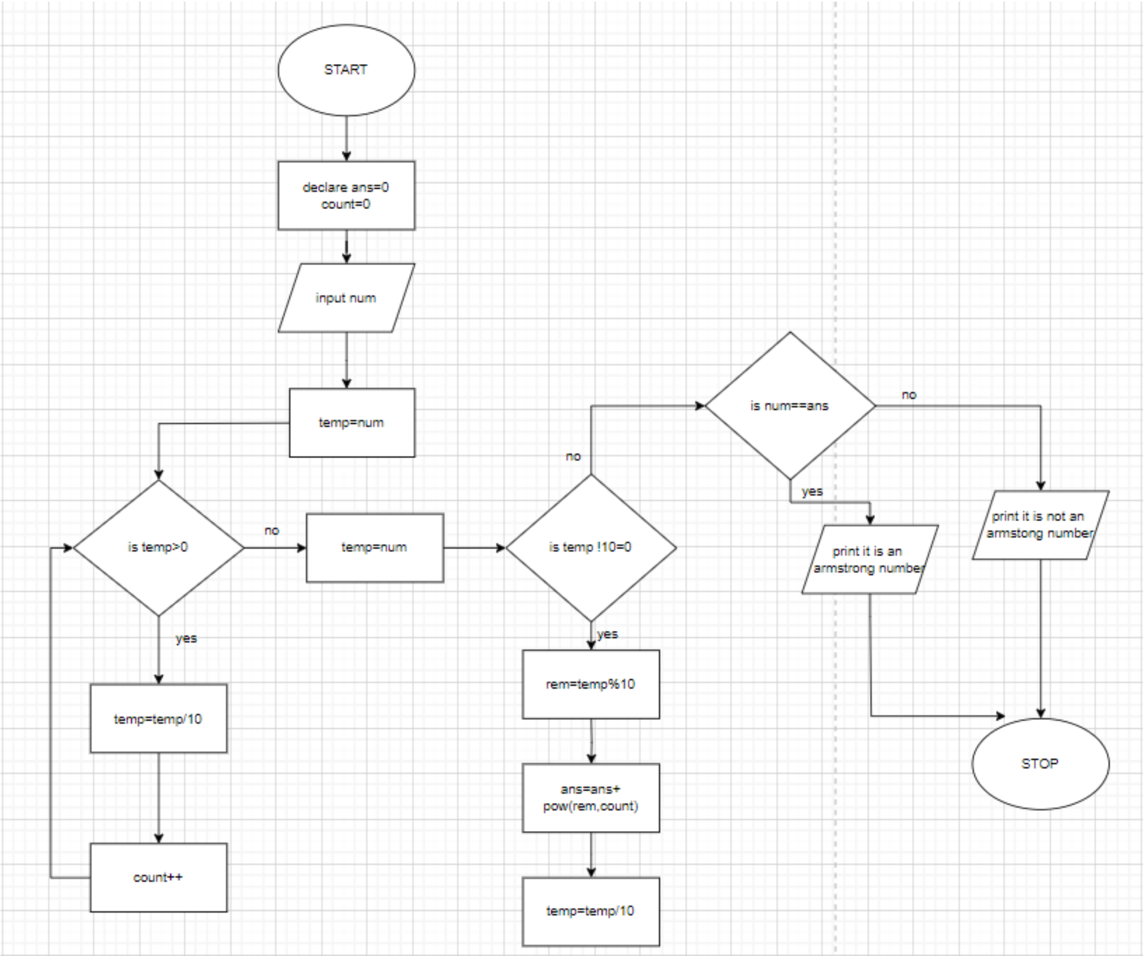
```
C:\Users\lekhn\OneDrive\Desktop\psipl>gcc area.c -o area

C:\Users\lekhn\OneDrive\Desktop\psipl>area
Menu
1.Circle
2.Rectangle
3.Triangle
Enter a number:1
Enter Radius:5
Area of circle is 78.500
C:\Users\lekhn\OneDrive\Desktop\psipl>area
Menu
1.Circle
2.Rectangle
3.Triangle
Enter a number:2
Enter Length:5
Enter Bredth:6
Area of Rectangle is 30.000
C:\Users\lekhn\OneDrive\Desktop\psipl>area
Menu
1.Circle
2.Rectangle
3.Triangle
Enter a number:3
Enter Height:5
Enter Base:6
Area of Triangle is 15.000
C:\Users\lekhn\OneDrive\Desktop\psipl>
```

Program 3

PROBLEM STATEMENT:	Write a program to find whether a number is armstrong or not using while loop.
ALGORITHM:	1. Start the program. 2. Declare integer variables num, count, arm, rem 3. Display the message "Enter Number for checking the length of num:" to prompt the user to enter a number. 4. Read the user's input into the num variable using scanf

5. Initialize the temp variable with the value of num to temporarily store the original number.
6. Use a while loop to count the number of digits in the num ; Initialize count = 0; While temp is greater than 0, divide it by 10, and increment count by 1.
7. Display the length of the number using printf
8. Reset the temp variable to the original num value.
9. Use another while loop to calculate the Armstrong number; Initialize arm to 0. While temp is greater than 0, calculate the remainder of temp divided by 10 and store it in rem. Add rem raised to the power of count to the arm variable. Divide temp by 10.
10. Display the value of arm using printf
11. Check if arm is equal to num.If they are equal, print "The number is an Armstrong number!", Otherwise, print "The number is not an Armstrong number."
12. End the program with a return value of 0.

FLOWCHART:

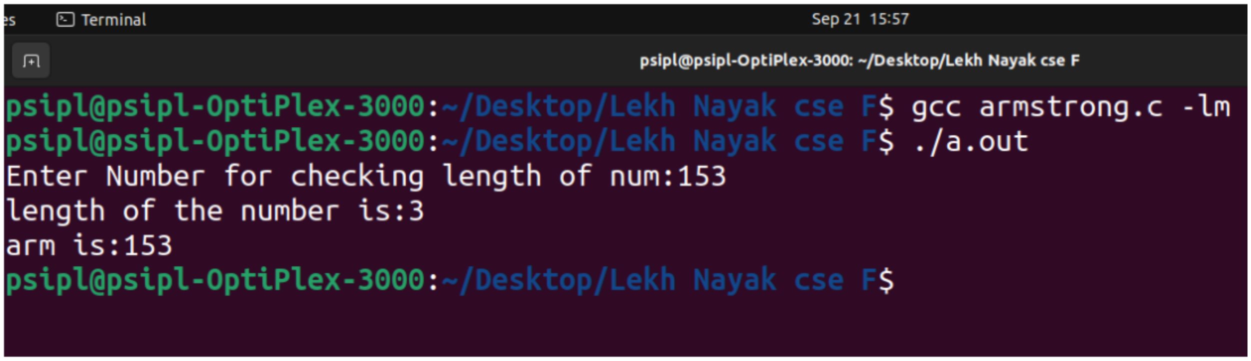


PROGRAM:

```
#include<stdio.h>
#include<math.h>
int main(){
```

```c
int num, count=0, arm=0,rem=0,temp=0;
  printf("Enter Number for checking length of num:");
  scanf("%d",&num);
temp=num;
while(temp>0){
       temp=temp/10;
       count++;
}
  printf("length of the number is:%d",count);
temp=num;
while(temp>0){
        rem=temp%10;
        arm=arm+pow(rem,count);
        temp=temp/10;
}
  printf("\narm is:%d",arm);

if(arm==num){
  printf("\nThe number is an armstrong number!");
}
else{
  printf("\nthe number is not an armstrong number");
}
return 0;
}
```

RESULT:



```
psipl@psipl-OptiPlex-3000:~/Desktop/Lekh Nayak cse F$ gcc armstrong.c -lm
psipl@psipl-OptiPlex-3000:~/Desktop/Lekh Nayak cse F$ ./a.out
Enter Number for checking length of num:153
length of the number is:3
arm is:153
psipl@psipl-OptiPlex-3000:~/Desktop/Lekh Nayak cse F$
```

Program 4

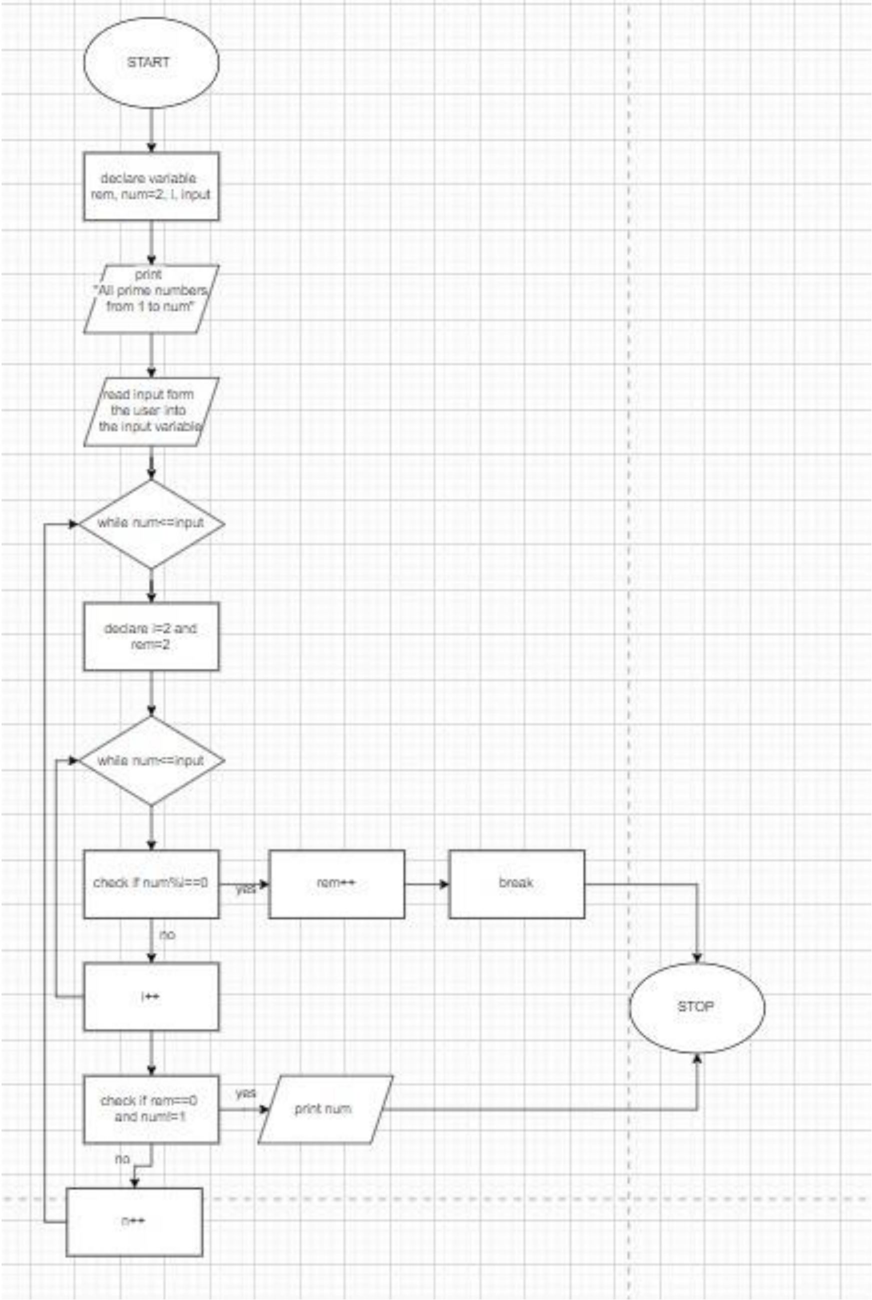
PROBLEM STATEMENT: Write a program to find the prime numbers within a range using for loop

ALGORITHM:

1. Start the program.
2. Declare variables rem, num, i, and input
3. Display the message "All prime numbers from 1 to " and get user input for upper limit and store it in variable input
4. Start a while loop that iterates from num (initially set to 2) up to input
5. Inside the loop, initialize i to 2 and reset rem to 0. These variables will be used to

	check for prime numbers. 6. Start another nested while loop that iterates from i to num/2. 7. In the nested loop, check if num is divisible by i (i.e., num%i=0).If it is divisible, increment the rem variable by 1 and break out of the loop using the break statement. 8. After the nested loop, check if rem is equal to 0 (meaning the number has no divisors other than 1 and itself) and if num is not equal to 1 (to exclude 1 from the prime numbers). 9. If the conditions in step 9 are met, print num, indicating that it's a prime number. 10. Increment num by 1 to move on to the next number. 11. Repeat steps 6-11 for all numbers from 2 to input. 12. Stop

FLOWCHART:

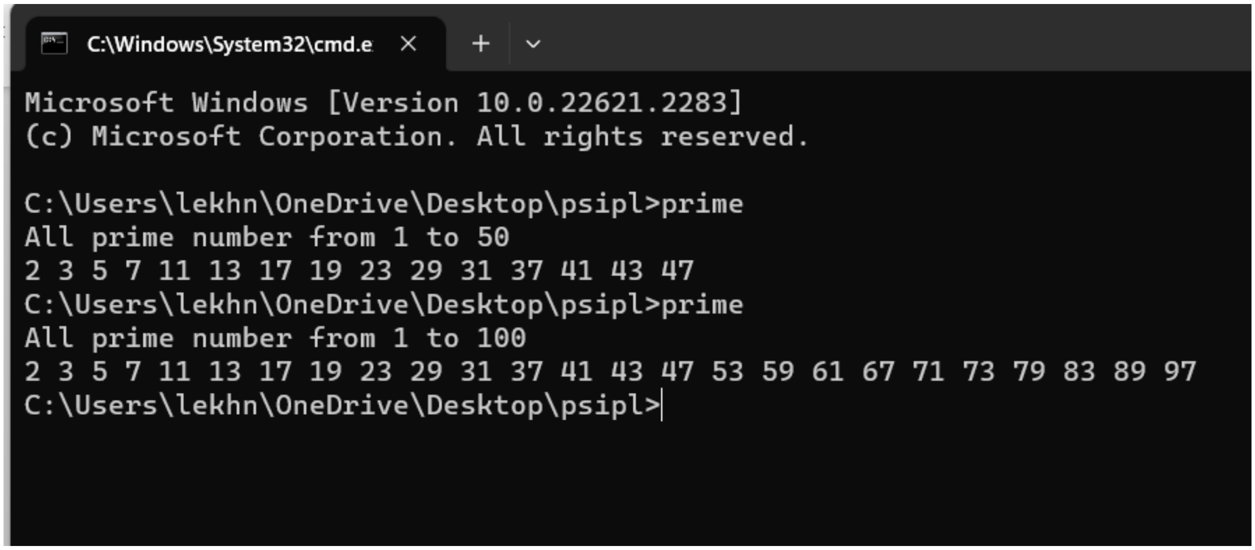

START
declare variable rem, num=2, i, input
print "All prime numbers from 1 to num"
read input form the user into the input variable
while num<=input
declare i=2 and rem=2
while num<=input
check if num%i==0
yes
rem++
break
no
i++
STOP
check if rem==0 and num!=1
yes
print num
no
n++

PROGRAM:

```
#include <stdio.h>
int main() {
        int rem = 0;
        int num = 2;
        int i =0;
        int input;
            printf("All prime number from 1 to ");
            scanf("%d",&input);
```

```
    while (num <= input) {
          i = 2;
          rem=0;
        while (i <= num/2) {
            if (num % i == 0) {
                rem++;
                break;
            }
            i++;
        }
        if (rem == 0 && num != 1) {
            printf("%d ", num);
        }
        num++;
    }
    return 0;
}
```

RESULT:



```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lekhn\OneDrive\Desktop\psipl>prime
All prime number from 1 to 50
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
C:\Users\lekhn\OneDrive\Desktop\psipl>prime
All prime number from 1 to 100
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
C:\Users\lekhn\OneDrive\Desktop\psipl>
```

Program 5

PROBLEM STATEMENT:

Write a program to print the following patterns(Take number of rows from the user):

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

 *
```

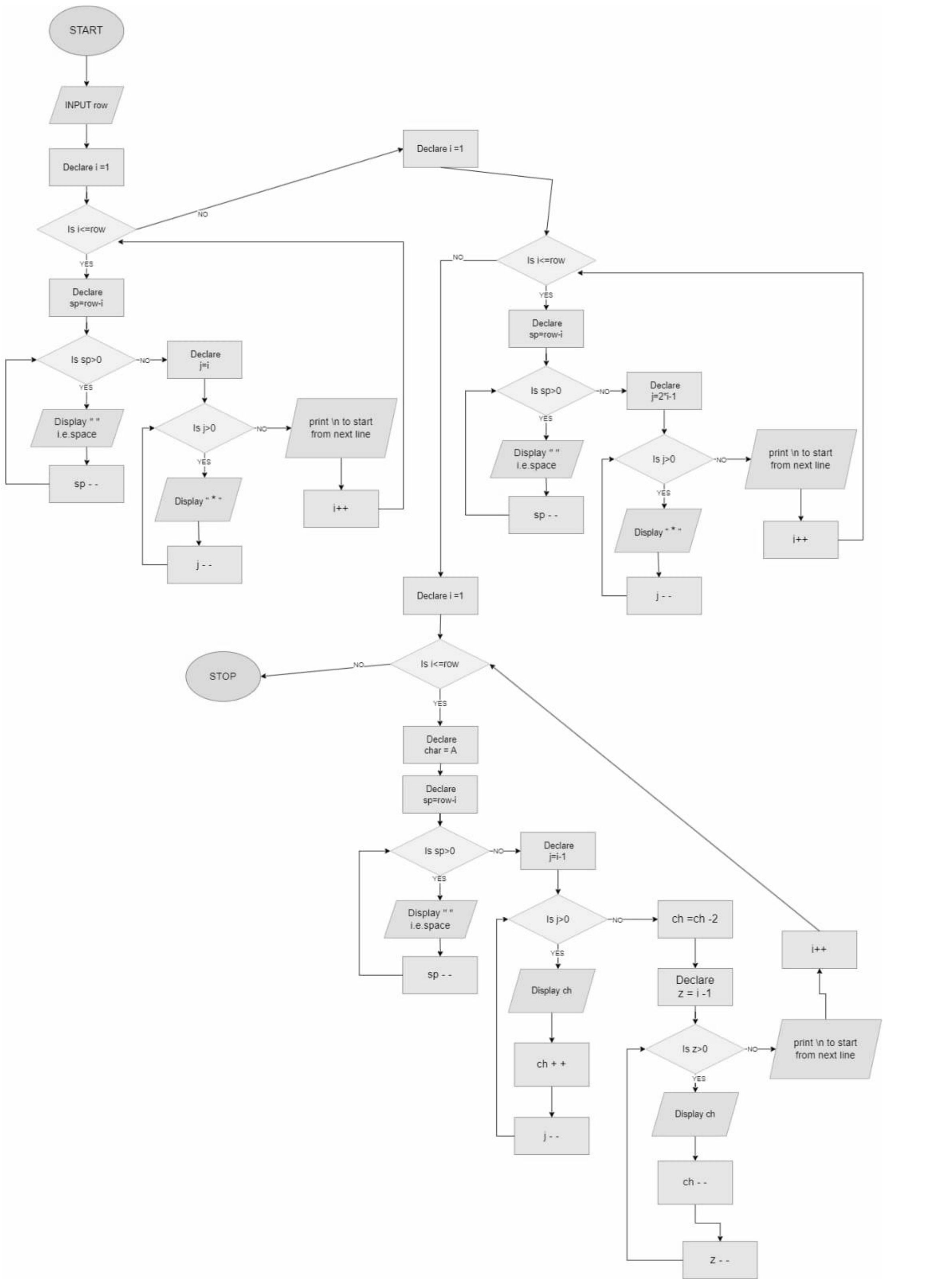
```
   * * *
  * * * * *
 * * * * * * *
* * * * * * * * *


      A
    A  B A
   A B C B A
  A B C  D C B A
 A B C  D E D C B A
```

ALGORITHM:

```
1.  Start the program.
2.  Declare an integer variable r to store the user input for the number of rows.
3.  Display the message "Enter number of rows:" to prompt the user to enter the number
    of rows.
4.  Read the user's input into the r variable using scanf.
5.  Print a blank line to separate the input prompt and the patterns.
6.  Pattern 1 (Left-aligned stars):
    6.1. Use a for loop with variable j to iterate from 1 to r (for each row).
    6.2. Inside the outer loop, use another for loop with variable i to print * characters i
         times.
    6.3. After the inner loop, print a newline character to move to the next row.
7.  Print two blank lines to separate Pattern 1 and Pattern 2.
8.  Pattern 2 (Right-aligned stars):
    8.1. Use a for loop with variable j to iterate from 1 to r (for each row).
    8.2. Inside the outer loop, use a for loop with variable sp to print spaces (r - j spaces)
         for alignment.
    8.3. After the space loop, use another for loop with variable i to print * characters i
         times.
    8.4. After the inner loops, print a newline character to move to the next row.
9.  Print two blank lines to separate Pattern 2 and Pattern 3.
10. Pattern 3 (Centered stars or triangle):
    10.1.     Use a for loop with variable j to iterate from 1 to r (for each row).
    10.2.     Inside the outer loop, use a for loop with variable sp to print spaces (r - j
         spaces) for alignment.
    10.3.     After the space loop, use another for loop with variable i to print *
         characters (2 * j - 1) times for a centered triangle.
    10.4.     After the inner loops, print a newline character to move to the next row.
11. Print two blank lines to separate Pattern 3 and Pattern 4.
12. Pattern 4 (Alphabetic characters - left and right triangles):
```

	12.1. Use a for loop with variable j to iterate from 1 to r (for each row). 12.2. Inside the outer loop, use a for loop with variable sp to print spaces (r - j spaces) for alignment. 12.3. After the space loop, use a character variable c initialized to 'A'. 12.4. Use another for loop with variable i to print characters in ascending order (c incremented) for the left triangle. 12.5. Decrease c by 2. 12.6. Use another for loop with variable i to print characters in descending order (c decremented) for the right triangle. 12.7. After the inner loops, print a newline character to move to the next row. 13. Stop

FLOWCHART:



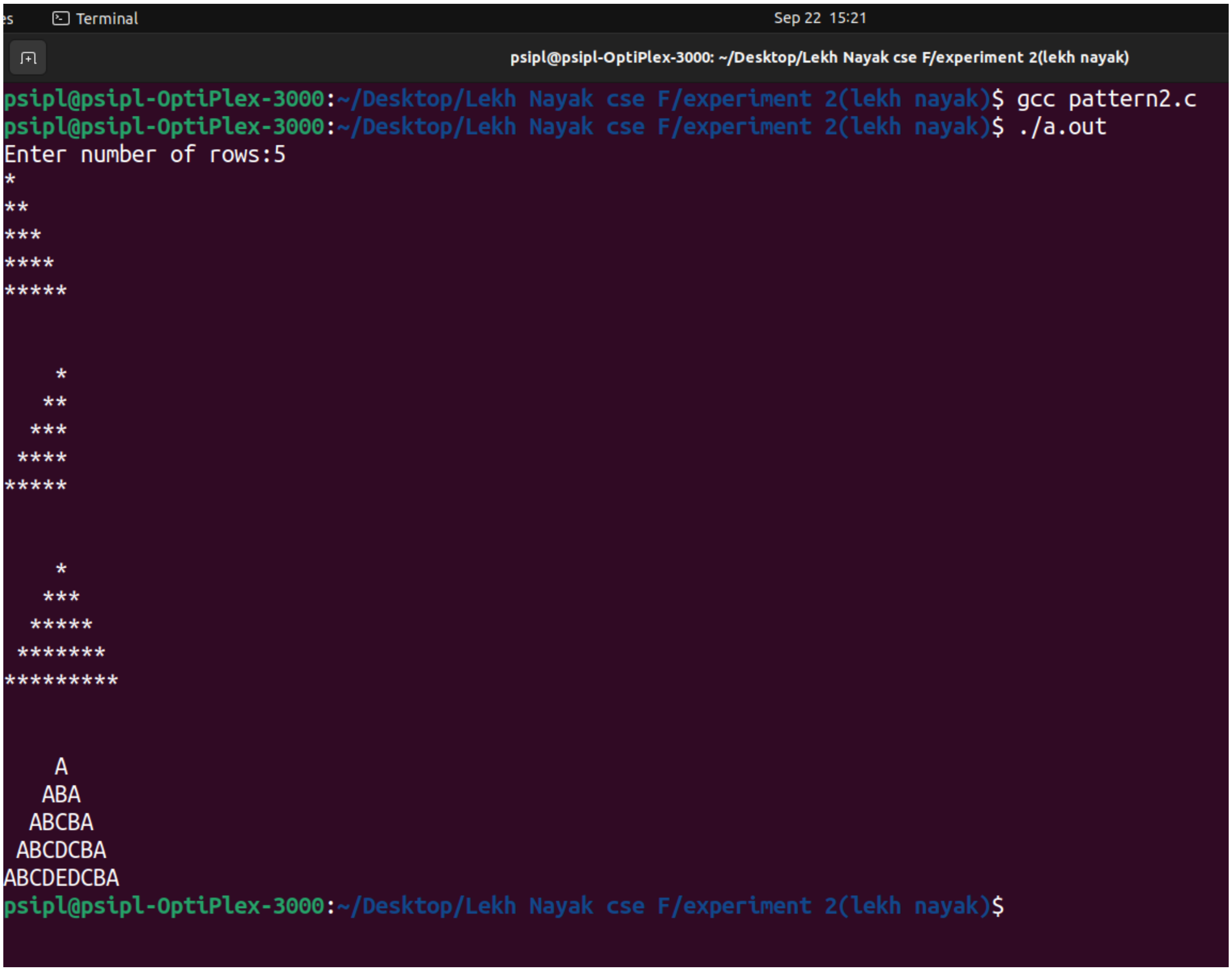
PROGRAM:

```
#include<stdio.h>
        int main(){
```

```
int r;
printf("Enter number of rows:");
scanf("%d",&r);
for(int j=1;j<=r;j++)//row loop
{
        for(int i=1;i<=j;i++)//star loop
          printf("*");
          printf("\n");

}

 printf("\n\n");

for(int j=1;j<=r;j++)//row loop
{
        for(int sp=1;sp<=r-j;sp++)//space loop
          printf(" ");
        for(int i=1;i<=j;i++)//star loop
          printf("*");
          printf("\n");
}

 printf("\n\n");

for(int j=1;j<=r;j++)//row loop
{
        for(int sp=1;sp<=r-j;sp++)//space loop
          printf(" ");
        for(int i=1;i<=2*j-1;i++)//star loop or triangle loop
          printf("*");
          printf("\n");
}
 printf("\n\n");

for(int j=1;j<=r;j++)//row loop
{
        for(int sp=1;sp<=r-j;sp++)//space loop
          printf(" ");
               char c='A';
        for(int i=1;i<=j;i++)//left triangle loop
        {
          printf("%c",c);
          c++;
        }
        c-=2;
        for(int i=1;i<=j-1;i++)//right triangle loop
        {
          printf("%c",c);
          c--;
        }
        printf("\n");
}
```

```
return 0;
}
```

RESULT:



s
Terminal
Sep 22 15:21
psipl@psipl-OptiPlex-3000: ~/Desktop/Lekh Nayak cse F/experiment 2(lekh nayak)

```
psipl@psipl-OptiPlex-3000:~/Desktop/Lekh Nayak cse F/experiment 2(lekh nayak)$ gcc pattern2.c
psipl@psipl-OptiPlex-3000:~/Desktop/Lekh Nayak cse F/experiment 2(lekh nayak)$ ./a.out
Enter number of rows:5
*
**
***
****
*****


    *
   **
  ***
 ****
*****


    *
   ***
  *****
 *******
*********


    A
   ABA
  ABCBA
 ABCDCBA
ABCDEDCBA
psipl@psipl-OptiPlex-3000:~/Desktop/Lekh Nayak cse F/experiment 2(lekh nayak)$
```

CONCLUSION:	In this experiment I learnt about control structures like if-else, switch, break, for, do-while and its application in problem solving.