**Evaluation 2**

**LOOPING CONTROL STRUCTURES-WHILE & DO LOOPS- Lab 3**

*1.Reverse a given number and check if it is a palindrome or not. (use while loop). [Ex: 1234, reverse=4\*10 3 +3 \* 10 2 + 2 \* 10 1 + 1 \* 10 0 =4321]*

**Program:**

#include <stdio.h>

int main()

{

int num,dig,rev=0;

printf("My name is Lajith Puthuchery and Registration Number is 200905106\n");

printf("Enter the number\n");

scanf("%d",&num);

int org = num;

while(num)

{

dig = num%10;

rev = rev\*10 + dig;

num = num/10;

}

printf("The reversed number is %d\n",rev);

if(org == rev)

{

printf("%d is a palindrome number",org);

}

else

{

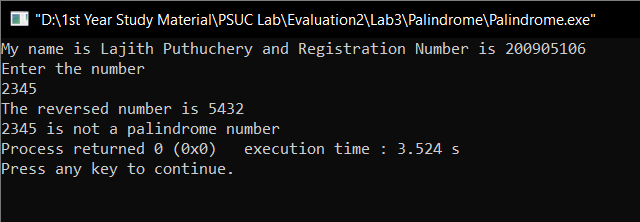
printf("%d is not a palindrome number",org);

}

return 0;

}

**Output:**

****

*2. Generate prime numbers between 2 given limits. (Use while loop)*

**Program:**

#include <stdio.h>

int main()

{

int a,b,j,flag=0;

printf("My name is Lajith Puthuchery and Registration Number is 200905106\n");

printf("Enter the limits between which the prime numbers need to be generated\n");

scanf("%d %d",&a,&b);

while(a<(b-1))

{

flag=0;

a++;

j=2;

while(j < a)

{

if((a)%j == 0)

{

flag=1;

break;

}

j++;

}

if(flag==0)

{

printf("%d ",a);

}

}

}

**Output:**

***Text

Description automatically generated***

*3. Check if the sum of the cubes of all digits of an inputted number equals the number itself (Armstrong Number). (Use while loop)*

**Program:**

#include <stdio.h>

#include <math.h>

int main()

{

int num,org,sum=0,dig;

printf("My name is Lajith Puthuchery and Registration Number is 200905106\n");

printf("Enter a number\n");

scanf("%d",&num);

org=num;

while(num)

{

dig = num%10;

sum += pow(dig,3);

num = num/10;

}

if(sum == org)

{

printf("%d is an Armstrong Number",org);

}

else

{

printf("%d is not an Armstrong Number",org);

}

return 0;

}

**Output:**

**Text

Description automatically generated**

*4. Write a program using do-while loop to read the numbers until -1 is encountered. Also count the number of prime numbers and composite numbers entered by user. [Hint: 1 is neither prime nor composite]*

**Program:**

#include <stdio.h>

int main()

{

int num;

int prime=0,comp=0;

int flag=0;

printf("My name is Lajith Puthuchery and Registration Number is 200905106\n");

do

{

printf("Enter a number");

scanf("%d",&num);

flag=0;

if(num==1)

{

continue;

}

if(num==-1)

{

break;

}

int j=2;

while(j<num)

{

if(num%j==0)

{

comp++;

flag=1;

break;

}

j++;

}

if(flag==0)

{

prime++;

}

}while(num);

printf("The number of prime numbers are %d\n",prime);

printf("The number of composite numbers are %d\n",comp);

return 0;

}

**Output:**

**A computer screen capture

Description automatically generated with low confidence**

*5. Check whether the given number is strong or not. [Hint: Positive number whose sum of the factorial of its digits is equal to the number itself] Ex: 145 = 1! + 4! + 5! = 1 + 24 + 120 = 145 is a strong number.*

**Program:**

#include <stdio.h>

int main()

{

int num,sum=0,dig,fact=1;

printf("My name is Lajith Puthuchery and Registration Number is 200905106\n");

printf("Enter a number\n");

scanf("%d",&num);

int org = num;

while(num)

{

dig = num%10;

fact =1;

while(dig!=1)

{

fact \*= dig;

dig--;

}

sum += fact;

num = num/10;

}

if(sum == org)

{

printf("%d is a Strong Number",org);

}

else

{

printf("%d is not a Strong Number",org);

}

return 0;

}

**Output:**

**Text

Description automatically generated**

*6. Write a program to demonstrate use of break and continue statements in while and do-while loops.*

**Program:**

#include <stdio.h>

int main()

{

int n=1;

printf("My name is Lajith Puthuchery and Registration Number is 200905106\n");

printf("While Loop\n");

printf("Break Statement\n");

while(n!=10)

{

if(n==5)

{

break;

}

printf("\t%d",n);

n++;

}

printf("\nContinue Statement\n");

n=1;

while(n!=10)

{

if(n==5)

{

n++;

continue;

}

printf("\t%d",n);

n++;

}

printf("\nDo While Loop\n");

printf("Break Statement\n");

n=1;

do

{

if(n==5)

{

break;

}

printf("\t%d",n);

n++;

} while(n!=10);

printf("\nContinue Statement\n");

n=1;

do

{

if(n==5)

{

n++;

continue;

}

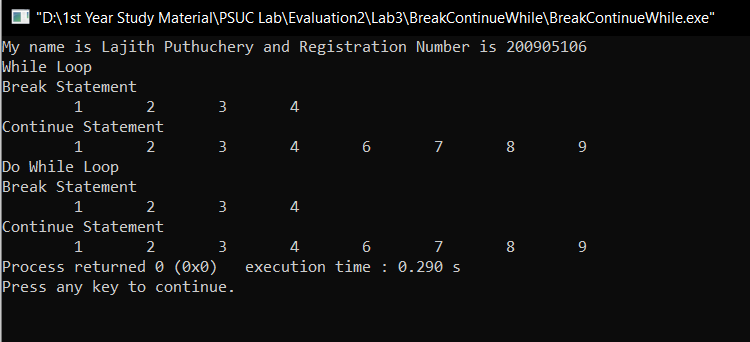
printf("\t%d",n);

n++;

} while(n!=10);

}

**Output:**

****

**LOOPING CONTROL STRUCTURES- FOR LOOPS – Lab 4**

*1.Generate the multiplication table for ‘n’ numbers up to ‘k’ terms (using nested for loops). [ Hint: 1 2 3 4 5 …. k 2 4 6 8 10 ….2\*k ..………………..… n……………… n\*k ]*

**Program:**

#include <stdio.h>

int main()

{

int n,k;

printf("My name is Lajith Puthuchery and registration number is 200905106\n");

printf("Enter the number 'n' till which the multiplication table should be generated and the term 'k' till which the table should be generated\n");

scanf("%d%d",&n,&k);

for(int i=1; i<=k; i++)

{

for(int j=1; j<=n; j++)

{

printf("%d\*%d = %d \t",j,i,i\*j);

}

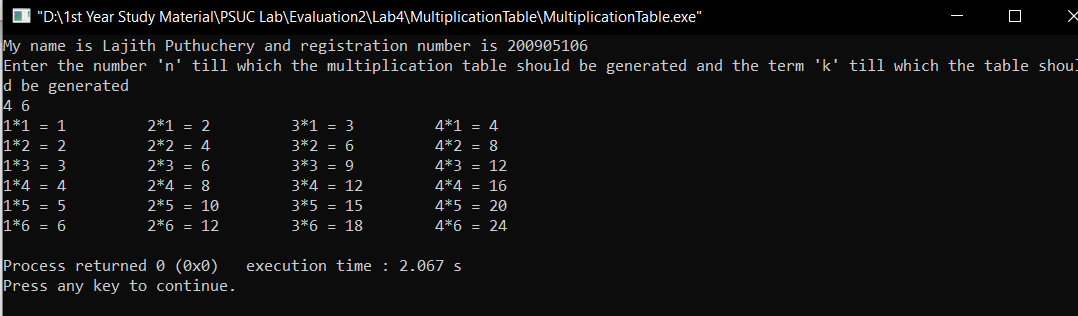
printf("\n");

}

return 0;

}

**Output:**

****

*2. Generate Floyd’s triangle using natural numbers for a given limit N. (using for loops) [Hint: Floyd’s triangle is a right angled-triangle using the natural numbers] Ex: Input: N = 4 14*

*Output: 1*

*2 3*

*4 5 6*

*7 8 9 10*

**Program:**

#include <stdio.h>

int main()

{

int n,m=1;

printf("My name is Lajith Puthuchery and registration number is 200905106\n");

printf("Enter the value for n for Floyd's Triangle\n");

scanf("%d",&n);

for(int i=1; i<=n; i++)

{

for(int j=1; j<=i ; j++)

{

printf("%d\t",m);

m++;

}

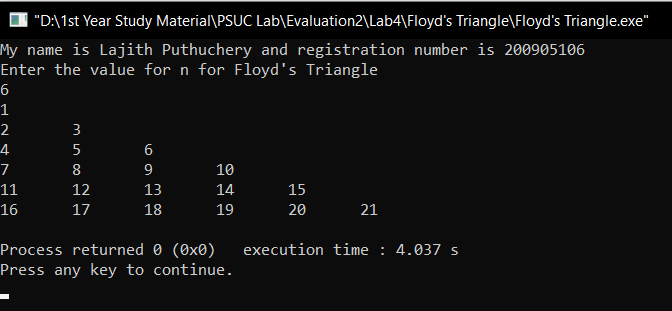
printf("\n");

}

return 0;

}

**Output:**

****

*3. Evaluate the sine series, sin(x)= x- x 3 /3! + x5 /5!–x 7 /7!+ ……… to n terms.*

**Program:**

#include <stdio.h>

int main()

{

float x,sum,t;

int n;

printf("My name is Lajith Puthuchery and registration number is 200905106\n");

printf("Enter the value of angle in degrees\n");

scanf("%f",&x);

printf("Enter the terms till which the series should be calculated");

scanf("%d",&n);

x=x\*3.14159/180;

t=x;

sum=x;

for(int i=1;i<=n;i++)

{

t=(t\*(-1)\*x\*x)/(2\*i\*(2\*i+1));

sum=sum+t;

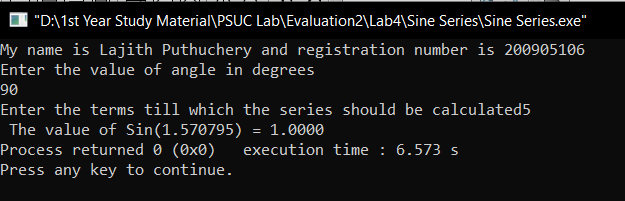
}

printf(" The value of Sin(%f) = %.4f",x,sum);

return 0;

}

**Output:**

****

*4. Check whether a given number is perfect or not. [Hint: Sum of all positive divisors of a given number excluding the given number is equal to the number] Ex: 28 = 1+ 2 + 4 + 7 + 14 = 28 is a perfect number*

**Program:**

#include <stdio.h>

int main()

{

int num,rem,sum=0;

printf("My name is Lajith Puthuchery and registration number is 200905106\n");

printf("Enter a number");

scanf("%d",&num);

for(int i=1; i<num; i++)

{

rem = num%i;

if(rem==0)

{

sum+= i;

}

}

if(sum==num)

{

printf("%d is a perfect number",num);

}

else

{

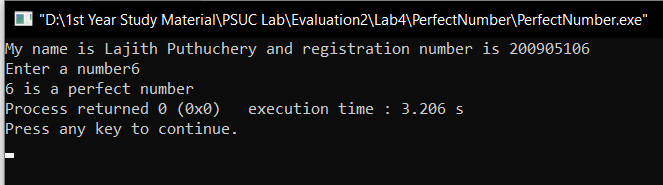
printf("%d is not a perfect number",num);

}

return 0;

}

**Output:**

****

*5. Find out the generic root of any number. [Hint: Generic root is the sum of digits of a number until a single digit is obtained.] Ex: Generic root of 456 is 4 + 5 + 6 = 15 = 1+5 = 6*

**Program:**

#include <stdio.h>

int main()

{

int num,dig,sum=0;

printf("My name is Lajith Puthuchery and registration number is 200905106\n");

printf("Enter a number whose generic root must be found\n");

scanf("%d",&num);

int orig = num;

while(num>10)

{

sum = 0;

while(num)

{

dig = num%10;

sum += dig;

num /= 10;

}

if(sum>10)

{

num=sum;

}

else

{

break;

}

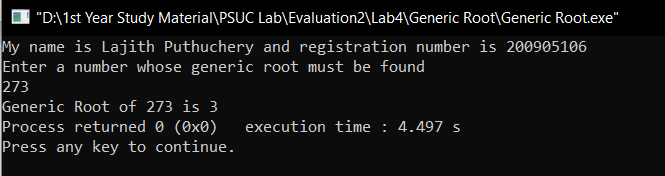
}

printf("Generic Root of %d is %d",orig,sum);

return 0;

}

**Output:**

****

*6. Write a program to demonstrate use of break and continue statements in for loop*

**Program:**

#include <stdio.h>

int main()

{

int i = 0, j = 0;

printf("My name is Lajith Puthuchery and Registration Number is 200905106\n");

printf("\nBreak Statement\n");

for (int i = 0; i < 5; i++)

{

printf("i = %d, j = ", i);

for (int j = 0; j < 5; j++)

{

//Break Statement

if (j == 2)

break;

printf("%d ", j);

}

printf("\n");

}

i = 0, j = 0;

printf("\nContinue Statement\n");

for (int i = 0; i < 5; i++)

{

printf("i = %d, j = ", i);

for (int j = 0; j < 5; j++)

{

// Continue Statement

if (j == 2)

continue;

printf("%d ", j);

}

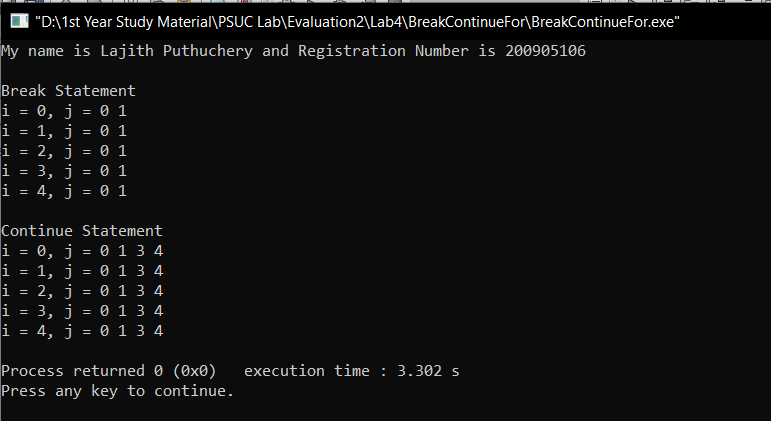
printf("\n");

}

return 0;

}

**Output:**

****