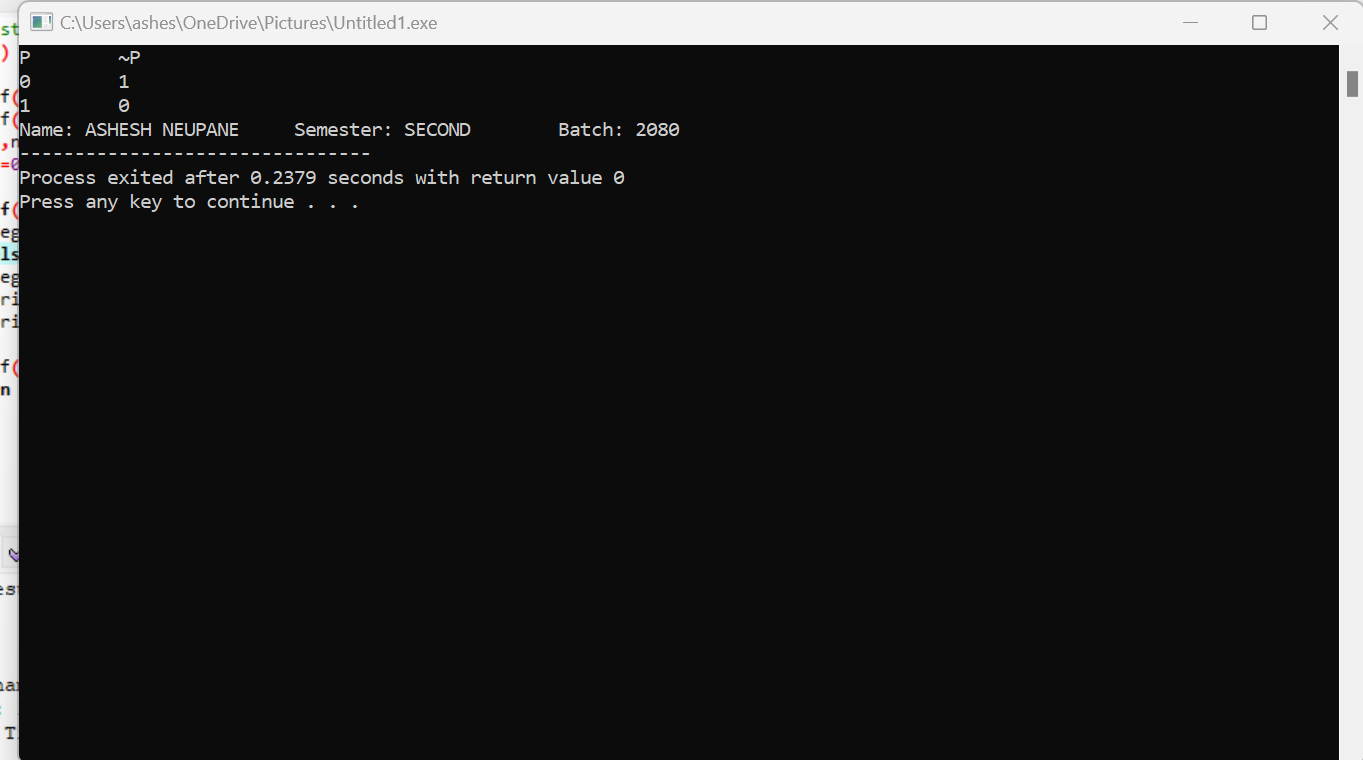
**(LAB – 1 : Digital Circuit)**

Write a program to print the truth table of Negation.

**Source Code:**

#include<stdio.h>  
int main()  
{  
 printf("P \t ~P");  
 printf("\n");  
 int p,negp;  
 for(p=0;p<=1;p++)  
 {  
 if(p==0)  
 negp=1;  
 else  
 negp=0;  
 printf("%d \t %d",p,negp);  
 printf("\n");  
 }  
 printf("Name: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
 return 0;  
}

**Output Screen:**

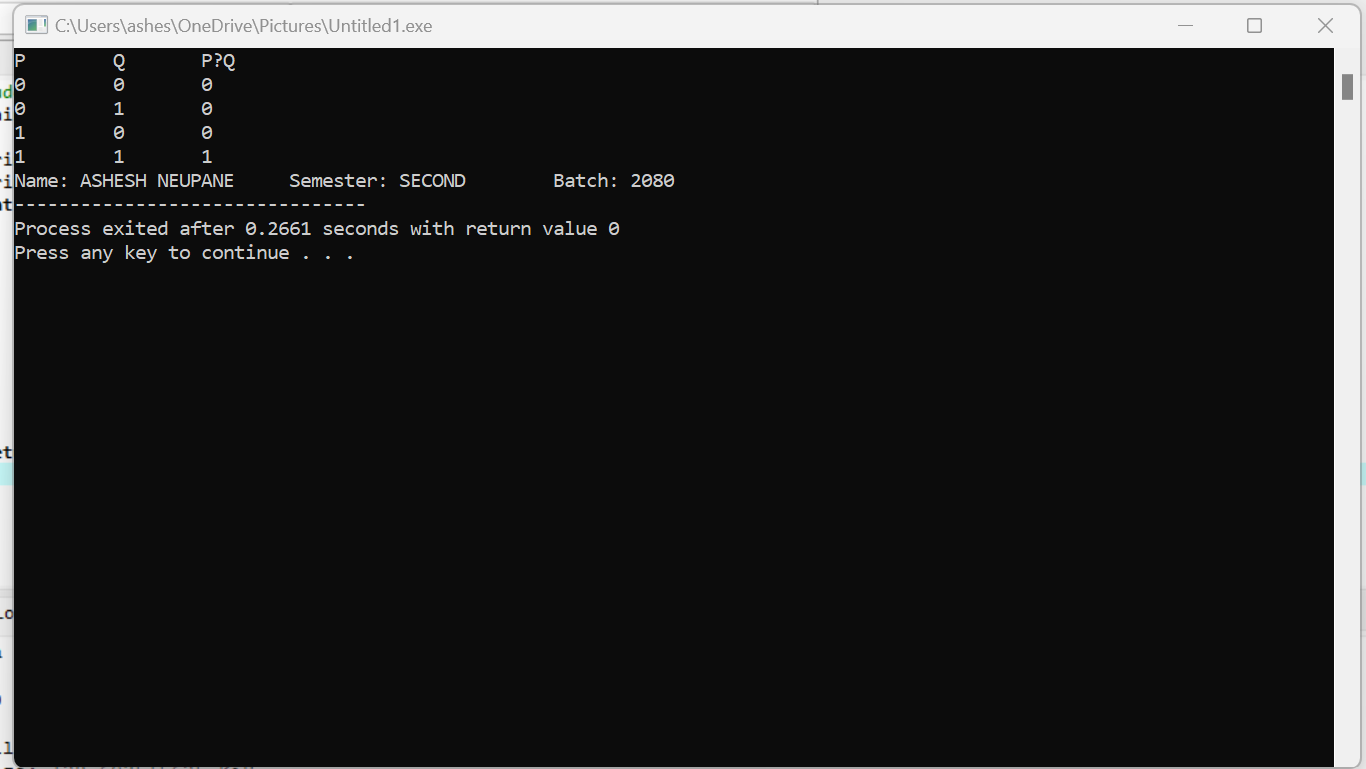


Write a program to print the truth table of conjunction.

**Source Code:**

#include<stdio.h>  
int main()  
{  
 printf("P \t Q \t P∧Q");  
 printf("\n");  
 int p,q,conjunction;  
 for(p=0;p<=1;p++)  
 {  
 for(q=0;q<=1;q++)  
 {  
 conjunction=p\*q;  
 printf("%d \t %d \t %d",p,q,conjunction);  
 printf("\n");  
 }  
 }  
 printf("Name: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
 return 0;  
}

**Output Screen:**



Write a program to print the truth table of disjunction.

**Source Code:**

#include<stdio.h>  
int main()  
{  
 printf("P \t Q \t P∨Q");  
 printf("\n");  
 int p,q,disjunction;  
 for(p=0;p<=1;p++)  
 {  
 for(q=0;q<=1;q++)  
 {  
 if(p==0 && q==0)  
 disjunction=0;  
 else  
 disjunction=1;  
 printf("%d \t %d \t %d",p,q,disjunction);  
 printf("\n");  
 }  
 }  
 printf("Name: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
 return 0;  
}

**Output Screen:**

**A screenshot of a computer

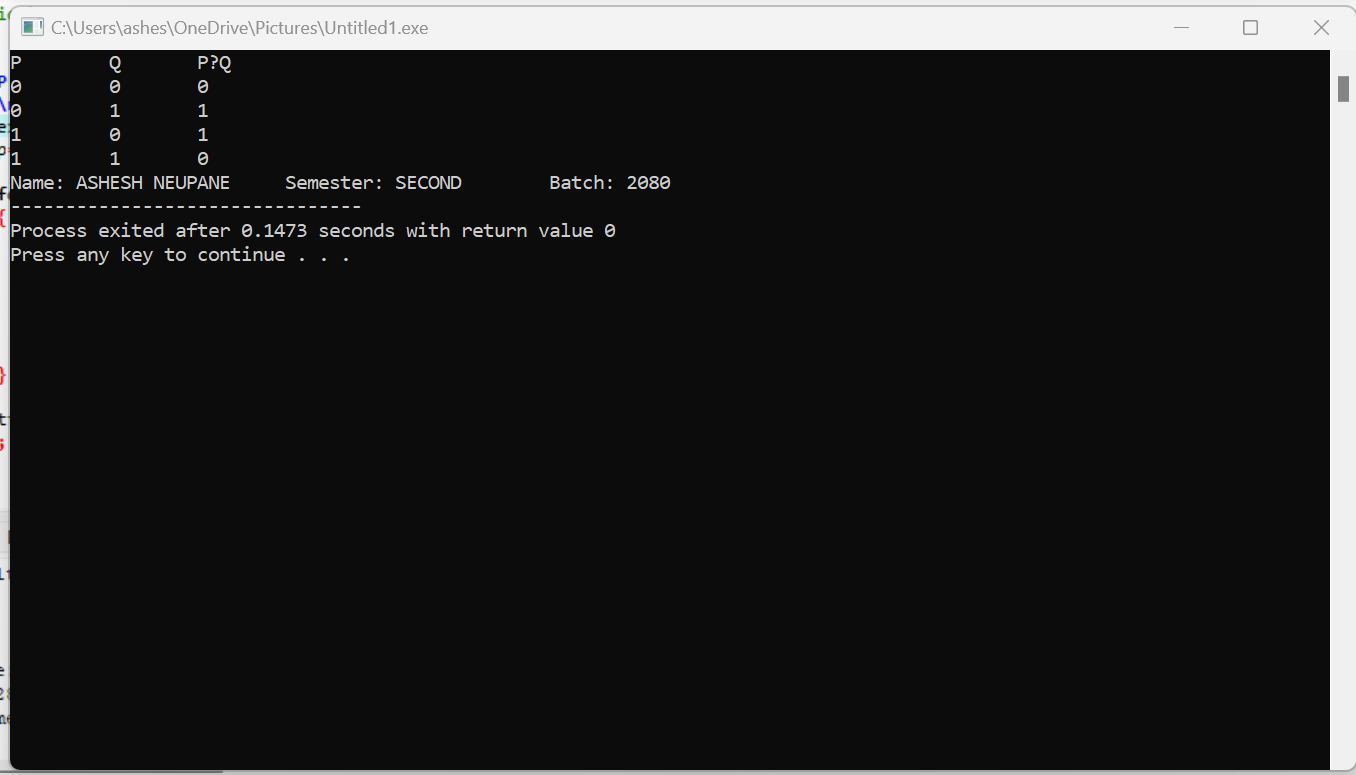
Description automatically generated**

Write a program to print truth table of Exclusive OR.

**Source Code:**

#include<stdio.h>  
int main()  
{  
 printf("P \t Q \t P⊕Q");  
 printf("\n");  
 int p,q,exclusiveor;  
 for(p=0;p<=1;p++)  
 {  
 for(q=0;q<=1;q++)  
 {  
 if(p==1&&q==1 || p==0 && q==0)  
 exclusiveor=0;  
 else  
 exclusiveor=1;  
 printf("%d \t %d \t %d",p,q,exclusiveor);  
 printf("\n");  
 }  
 }  
 printf("Name: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
 return 0;  
}

**Output Screen:**



Write a program to print truth table of Conditional.

**Source Code:**

#include<stdio.h>  
int main()  
{  
 printf("P \t Q \t P → Q ");  
 printf("\n");  
 int p,q,conditional;  
 for(p=0;p<=1;p++)  
 {  
 for(q=0;q<=1;q++)  
 {  
 if(p==1 && q==0)  
 conditional=0;  
 else  
 conditional=1;  
 printf("%d \t %d \t %d",p,q,conditional);  
 printf("\n");  
 }  
 }  
 printf("Name: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
 return 0;  
}

**Output Screen:**

A computer screen with white text

Description automatically generated

Write a program to print truth table of Bi - Conditional.

**Source Code:**

#include<stdio.h>  
int main()  
{  
 printf("P \t Q \t P ↔ Q");  
 printf("\n");  
 int p,q,biconditional;  
 for(p=0;p<=1;p++)  
 {  
 for(q=0;q<=1;q++)  
 {  
 if(p==1 && q==1 || p==0 && q==0)  
 biconditional=1;  
 else  
 biconditional=0;  
 printf("%d \t %d \t %d",p,q,biconditional);  
 printf("\n");  
 }  
 }  
 printf("Name: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
 return 0;  
}

**Output Screen:**

A screenshot of a computer

Description automatically generated

**(LAB – 2 : Finding GCD by using Euclidian Algorithm)**

Write a program to find GCD by using Euclidian Algorithm.

**Source Code:**

#include<stdio.h>  
int GCD(int a,int b)   
{  
 while (b!=0)   
 {  
 int temp=b;  
 b=a%b;  
 a=temp;  
 }  
 return a;  
}  
int main()   
{  
 int num1, num2, gcd;  
 printf("Enter two numbers to find GCD: ");  
 scanf("%d %d",&num1,&num2);  
 gcd = GCD(num1,num2);  
 printf("GCD of %d and %d is: %d\n",num1,num2,gcd);  
 printf("Name: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
 return 0;  
}

**Output Screen:**

**A screen shot of a computer

Description automatically generated**

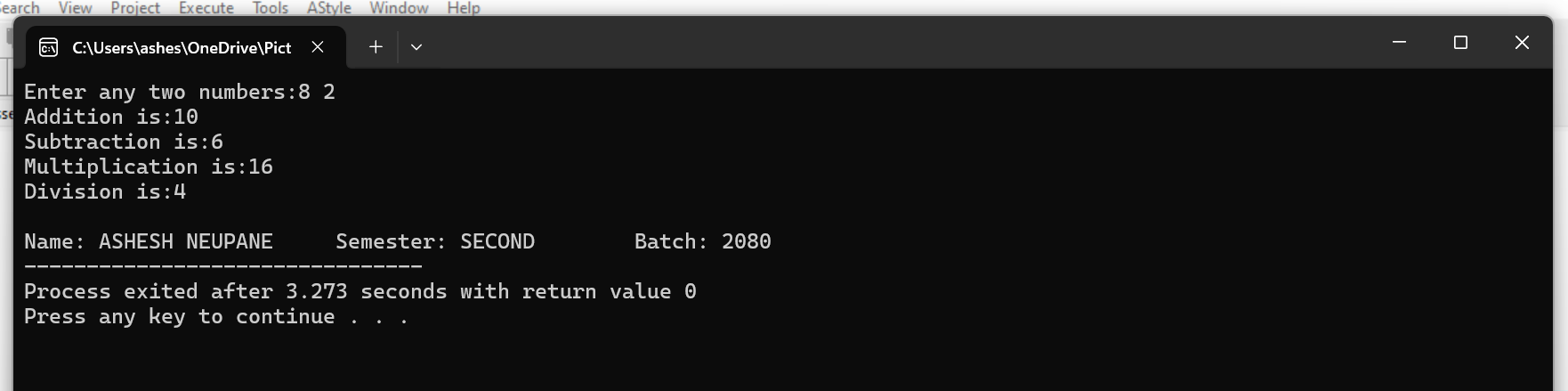
**(LAB – 3)**

Write a program to calculate addition, subtraction, multiplication and division of the two integer number.

**Source Code:**

#include<stdio.h>  
int main()  
{  
 int a,b;  
 printf("Enter any two numbers:");  
 scanf("%d%d",&a,&b);  
 printf("Addition is:%d\n",a+b);  
 printf("Subtraction is:%d\n",a-b);  
 printf("Multiplication is:%d\n",a\*b);  
 printf("Division is:%d\n",a/b);  
 printf("\n");  
 printf("Name: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");   
 return 0;  
}

**Output Screen:**



**(LAB – 4)**

Write a program to print the ceiling value and flooring value of the given number.

**Source Code:**

#include<stdio.h>  
#include<math.h>  
int main()  
{  
 float number;  
 printf("Enter a number:");  
 scanf("%f",&number);  
 printf("Ceiling Value is: %.2f\n",ceil(number));  
 printf("Floor Value is: %.2f\n",floor(number));  
 printf("Name: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");   
 return 0;  
}

**Output Screen:**

A screenshot of a computer

Description automatically generated

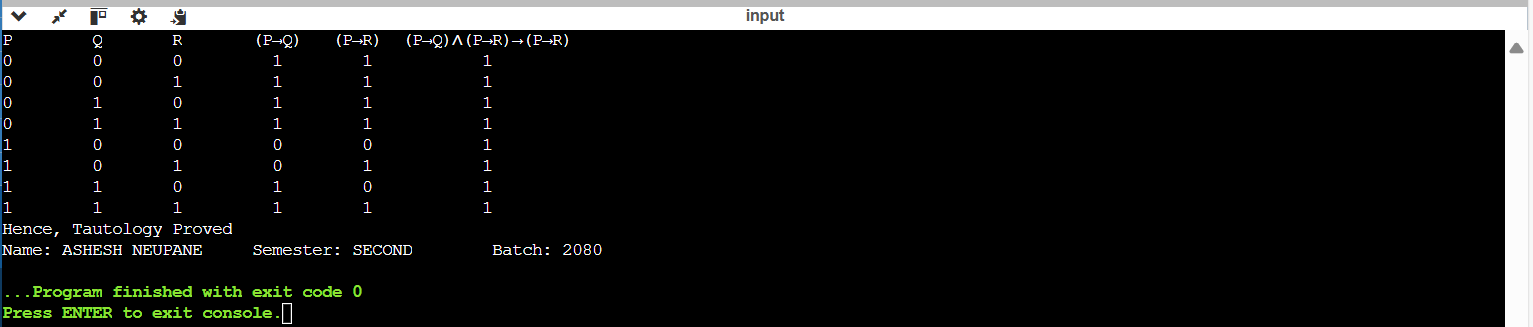
**(LAB – 5)**

Write a program to prove that (p→q) [∧ (p→r) → (p→r) is tautology.](https://www.piliapp.com/symbols/conjunction/)

**Source Code:**

#include<stdio.h>  
int main()  
{  
 int p,q,r,result1,result2,demo,result;  
 printf("P \t Q \t R \t (P→Q)\t (P→R)\t(P→Q)∧(P→R)→(P→R)");  
 printf("\n");  
 for(p=0;p<=1;p++)  
 {  
 for(q=0;q<=1;q++)  
 {  
 for(r=0;r<=1;r++)  
 {  
 if(p==1&&q==0)  
 result1=0;  
 else  
 result1=1;  
 if(p==1&&r==0)  
 result2=0;  
 else  
 result2=1;  
 if(result1==1&&result2==1)  
 demo=1;  
 else  
 demo=0;  
 if(demo==1&&result2==0)  
 result=0;  
 else  
 result=1;  
 printf("%d \t %d \t %d \t %d \t %d \t %d \n",p,q,r,result1,result2,result);  
 }  
 }  
 }  
 printf("Hence, Tautology Proved");  
 printf("\nName: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");   
 return 0;  
}

**Output Screen:**



**(LAB – 6)**

Write a program to find the factorial of a given number using recursion.

**Source Code:**

#include<stdio.h>  
int fact(int n);  
int main()   
{   
 int n;   
 printf("Enter a number: ");   
 scanf("%d",&n);   
 if (n < 0)   
 printf("Factorial cannot be determined");  
 else  
 printf("Factorial is: %d\n", fact(n));  
 printf("\nName: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
 return 0;  
}  
int fact(int n)   
{   
 if (n == 0)   
 return 1;   
 else   
 return n \* fact(n - 1);   
}

**Output Screen:**

A screen shot of a computer

Description automatically generated

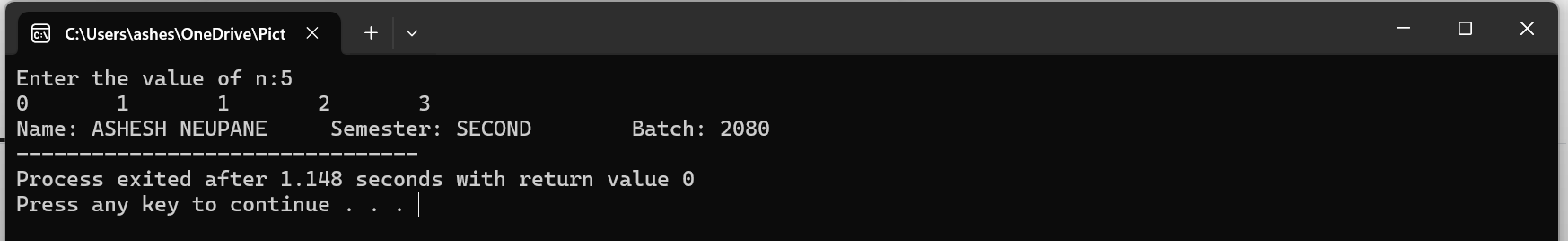
**(LAB – 7)**

Write a program to print Fibonacci series upto n using recursion.

**Source Code:**

#include<stdio.h>   
int fibo(int n);   
int main()   
{   
 int i,n;   
 printf("Enter the value of n:");  
 scanf("%d",&n);  
 for(i=0;i<n;i++)   
 {   
 int r=fibo(i);   
 printf("%d\t",r);   
 }   
 printf("\nName: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
}   
int fibo(int n)   
{   
 if(n==0)   
 return 0;   
 else if(n==1)   
 return 1;   
 else   
 return fibo(n-1)+fibo(n-2);   
}

**Output Screen:**



Write a program to print Fibonacci series upto n without using recursion.

**Source Code:**

#include<stdio.h>  
int main()   
{  
 int n,first=0,second=1,next;  
 printf("Enter the value of n: ");  
 scanf("%d",&n);  
 printf("Fibonacci series: ");  
 for (int i = 0; i < n; i++)   
 {  
 if (i == 0)   
 printf("%d ",first);  
 else if (i == 1)  
 printf("%d ", second);  
 else   
 {  
 next=first+second;  
 printf("%d ",next);  
 first=second;  
 second=next;  
 }  
 }  
 printf("\nName: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
 return 0;  
}

**Output Screen:**

A screen shot of a computer

Description automatically generated

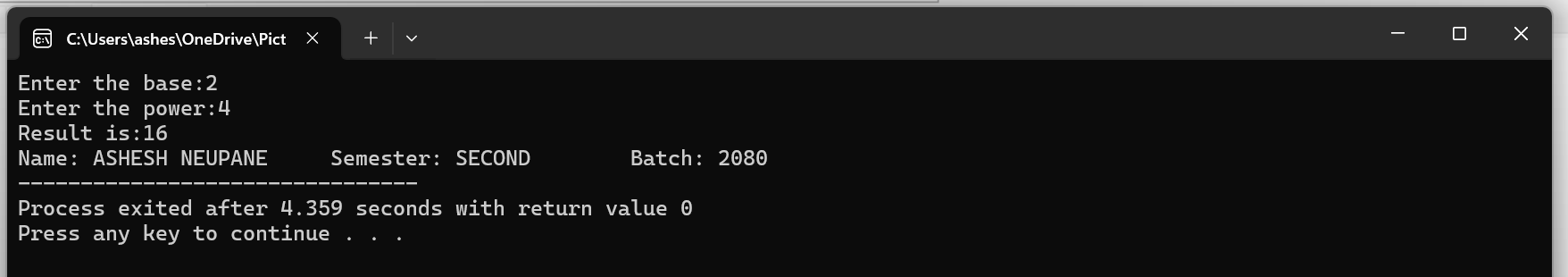
**(LAB – 8)**

Write a program to compute power of any number.

**Source Code:**

#include<stdio.h>  
#include<math.h>  
int main()  
{  
 int base,power,result;  
 printf("Enter the base:");  
 scanf("%d",&base);  
 printf("Enter the power:");  
 scanf("%d",&power);  
 result=pow(base,power);  
 printf("Result is:%d",result);  
 printf("\nName: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
}

**Output Screen:**

****

**(LAB – 9)**

Write a program to print the permutated string “ABC”.

**Source Code:**

#include <stdio.h>  
int printPermutations(char \*str, int length)   
{  
 char temp;  
 for (int i = 0; i < length; i++)   
 {  
 for (int j = 0; j < length; j++)   
 {  
 for (int k = 0; k < length; k++)   
 {  
 if (i != j && i != k && j != k)   
 printf("%c%c%c\n", str[i], str[j], str[k]);  
 }  
 }  
 }  
 printf("\nName: ASHESH NEUPANE \t Semester: SECOND \t Batch: 2080");  
}  
int main()   
{  
 char str[] = "ABC";  
 int length = sizeof(str) - 1;  
 printf("Permutations of the string \"ABC\":\n");  
 printPermutations(str, length);  
 return 0;  
}

**Output Screen:**

**A screen shot of a computer

Description automatically generated**