Experiment 1: Introduction to C++ and I/O

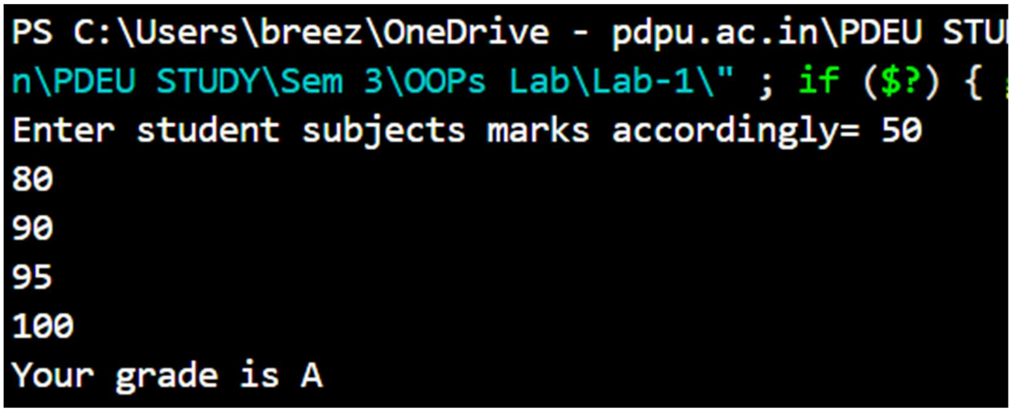
Problem 1: Write a C++ program to calculate and display the grade of a student based on marks in 5 subjects.

Code:

```
#include <iostream>
using namespace std;
int main() {
   int a,b,c,d,e;
   int sum;
   cout<<"Enter student subjects marks accordingly= ";
   cin>>a>>b>>c>>d>>e;
   sum=(a+b+c+d+e)/5;
   if(sum>=90)
   {
      cout<<"Your grade is A+";
   }
   else if (sum<90 && sum>=80)
   {
      cout<<"Your grade is A";
   }
   else if (sum<80 && sum>=70)
   {
      cout<<"Your grade is B+";
   }
   else if (sum<70 && sum>=60)
   {
      cout<<"Your grade is B";
   }
   else if (sum<60 && sum>=50)
```

```
    {
        cout<<"Your grade is C";
    }
    else
    {
        cout<<"Fail";
    }
    return 0;
}
```

Output:



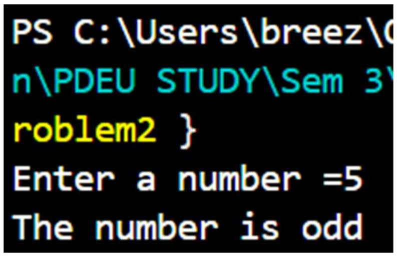
```
PS C:\Users\breez\OneDrive - pdpu.ac.in\PDEU STU
n\PDEU STUDY\Sem 3\OOPs Lab\Lab-1\" ; if ($?) {
Enter student subjects marks accordingly= 50
80
90
95
100
Your grade is A
```

Problem 2: Write C++ program to find whether number is even or odd.

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    cout<<"Enter a number =";
    cin>>a;
    if(a%2==0)
    {
       cout<<"The number is even";
    }
    else
    {
       cout<<"The number is odd";
    }
}
```

Output:



```
PS C:\Users\breez\
n\PDEU STUDY\Sem 3
roblem2 }
Enter a number =5
The number is odd
```

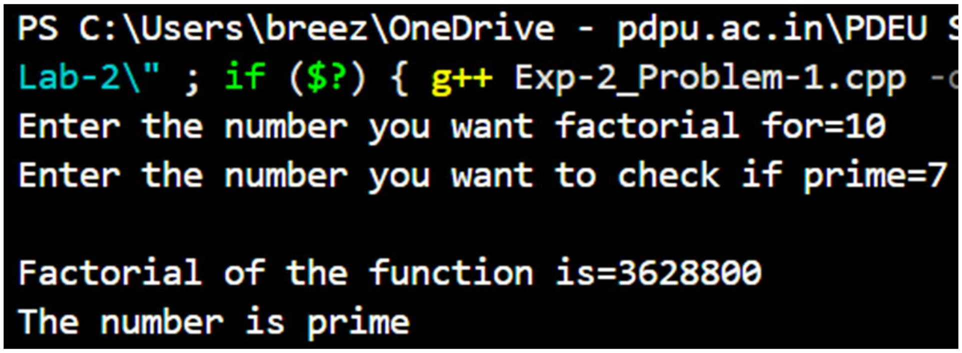
Experiment 2: Functions and Control Structures

Problem 1: Write a program in C++ using recursive function to compute factorial and check for prime numbers.

Code:

```
#include <iostream>
using namespace std;
int fact=1;
int factorial(int n)
{
    fact=fact*n;
    n=n-1;
    if (n!=0)
    {
        return factorial(n);
    }
    else
    {
        return fact;
    }
}

int prime(int n)
{
    int count=0;
    for(int i=1;i<=n;i++)
    {
        if(n%i==0)
        {
            count+=1;
        }
    }
    if(count==2)
    {
        cout<<"\nThe number is prime";
    }
    else
    {
        cout<<"\nThe number is not prime";
    }
    return 0;
}

int main()
{
    int n,p;
    cout<<"Enter the number you want factorial for=";
    cin>>n;
    cout<<"Enter the number you want to check if prime=";
    cin>>p;
    int b=factorial(n);
    cout<<"\nFactorial of the function is="<<b;
    prime(p);
    return 0;
}
```

Output:



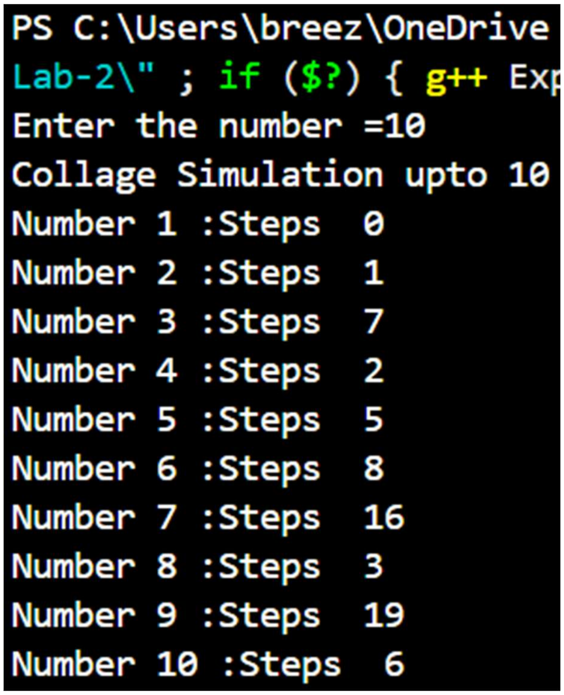
```
PS C:\Users\breez\OneDrive - pdpu.ac.in\PDEU S
Lab-2\" ; if ($?) { g++ Exp-2_Problem-1.cpp -
Enter the number you want factorial for=10
Enter the number you want to check if prime=7

Factorial of the function is=3628800
The number is prime
```

Problem 2: Simulate Collatz Conjecture for 1 to N and find the number with longest steps in C++.

Code:

```
#include <iostream>
using namespace std;
void conjecture(int a,int b)
{
   int count1=0;

   while(a!=1)
   {
      if(a%2==0)
      {
         a=a/2;
         count1+=1;
      }
      else
      {
         a=(3*a+1);
         count1+=1;
      }
   }
   cout<<"Number "<<b<<" "<<":"<<"Steps  "<<count1<<"\n";
}

int main()
{
   int a;
   cout<<"Enter the number =";
   cin>>a;
   cout<<"Collage Simulation upto "<<a<<"\n";
   for(int i=1;i<=a;i++)
   {
      conjecture(i,i);
   }
   return 0;
}
```

Output:



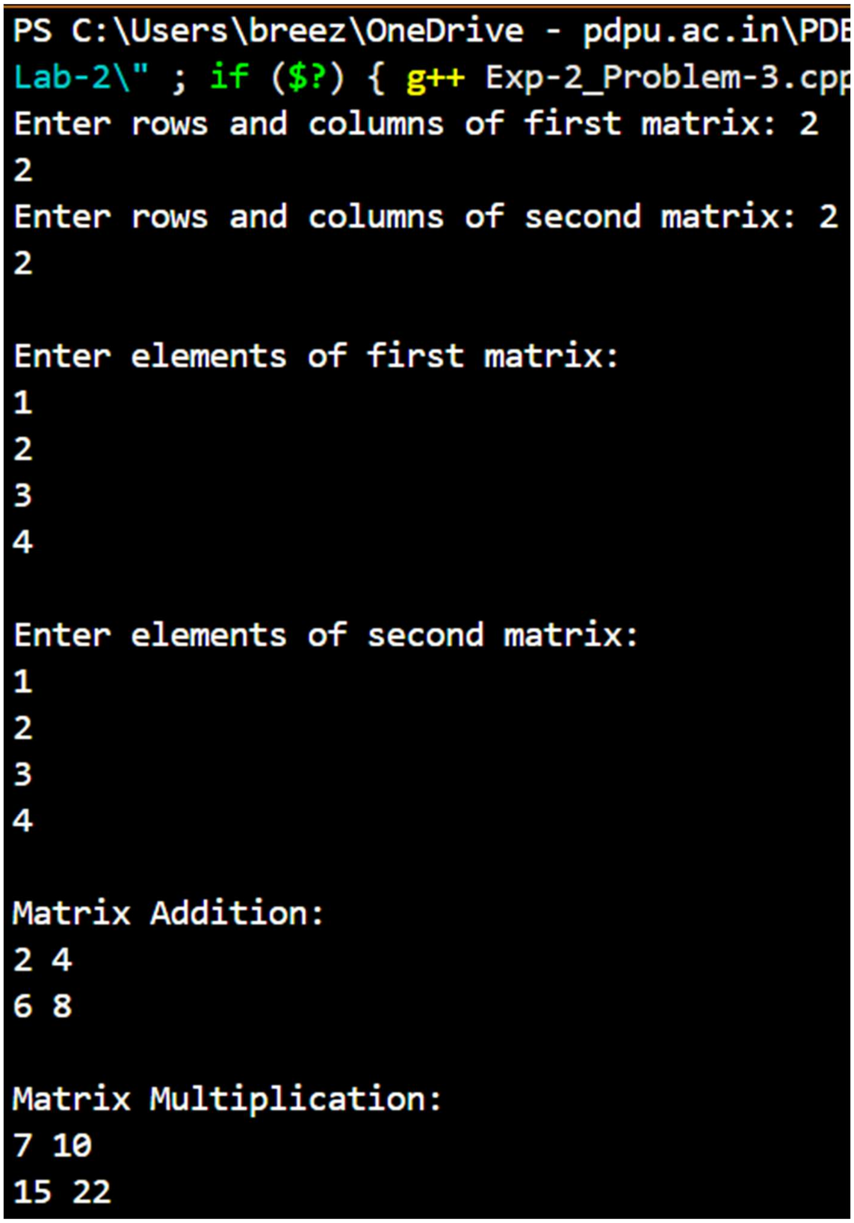
```
PS C:\Users\breez\OneDrive
Lab-2\" ; if ($?) { g++ Exp
Enter the number =10
Collage Simulation upto 10
Number 1 :Steps  0
Number 2 :Steps  1
Number 3 :Steps  7
Number 4 :Steps  2
Number 5 :Steps  5
Number 6 :Steps  8
Number 7 :Steps  16
Number 8 :Steps  3
Number 9 :Steps  19
Number 10 :Steps  6
```

Problem 3: Write a C++ program to perform addition and multiplication of Two Matrices.

Code:

```
#include <iostream>
using namespace std;

int main()
{
   int m,n,p,q;
   cout<<"Enter rows and columns of first matrix: ";
   cin>>m>>n;
   cout<<"Enter rows and columns of second matrix: ";
   cin>>p>>q;

   int A[100][100],B[100][100],sum[100][100],product[100][100];

   cout<<"\nEnter elements of first matrix:\n";
   for(int i=0;i<m;i++)
   {
      for(int j=0;j<n;j++)
      {
         cin>>A[i][j];
      }
   }

   cout<<"\nEnter elements of second matrix:\n";
   for(int i=0;i<p;i++)
   {
      for(int j=0;j<q;j++)
      {
        cin>>B[i][j];
    }
}

if(m==p && n==q)
{
    cout<<"\nMatrix Addition:\n";
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            sum[i][j]=A[i][j]+B[i][j];
            cout<<sum[i][j]<<" ";
        }
        cout<<"\n";
    }
}
else
{
    cout<<"\nMatrix addition is not possible.\n";
}

if(n==p)
{
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<q;j++)
        {
            product[i][j]=0;
            }
        }
        cout<<"\nMatrix Multiplication:\n";
        for(int i=0;i<m;i++)
        {
            for(int j=0;j<q;j++)
            {
                for(int k=0;k<n;k++)
                {
                    product[i][j]+=A[i][k]*B[k][j];
                }
                cout<<product[i][j]<<" ";
            }
            cout<<"\n";
        }
    }
    else
    {
        cout<<"\nMatrix multiplication is not possible.\n";
    }
    return 0;
}
```

Output:



```
PS C:\Users\breez\OneDrive - pdpu.ac.in\PDE
Lab-2\" ; if ($?) { g++ Exp-2_Problem-3.cpp
Enter rows and columns of first matrix: 2
2
Enter rows and columns of second matrix: 2
2

Enter elements of first matrix:
1
2
3
4

Enter elements of second matrix:
1
2
3
4

Matrix Addition:
2 4
6 8

Matrix Multiplication:
7 10
15 22
```

Experiment 3: Introduction to Object-Oriented Programming and Encapsulation

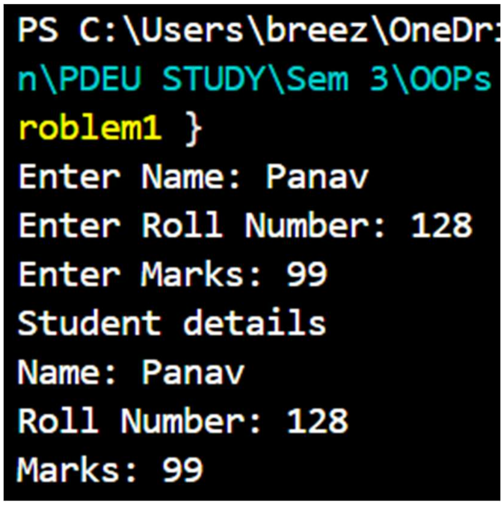
Problem 1: Create a class Student with the following:

- Private members: name, rollNumber, marks
- Public methods:
  - setDetails() – to set values
  - displayDetails() – to show student information

Code:

```
#include<iostream>
#include<cstring>
using namespace std;
class Student
{
 private:
   char name[100];
   int rollNumber;
   int marks;
 public:
   void setDetails(char n[],int r,int m)
   {
     strcpy(name,n);
     rollNumber=r;
     marks=m;
   }

   void displayDetails()
   {
     cout<<"Student details\n"<<"Name: "<<name<<"\n"<<"Roll Number: 
"<<rollNumber<<"\n"<<"Marks: "<<marks;
   }
};

int main()
{
 Student s1;
 char n[100];
 int r;
 int m;
 cout<<"Enter Name: ";
 cin>>n;
 cout<<"Enter Roll Number: ";
 cin>>r;
 cout<<"Enter Marks: ";
 cin>>m;
 s1.setDetails(n,r,m);
 s1.displayDetails();
 return 0;
}
```

Output:



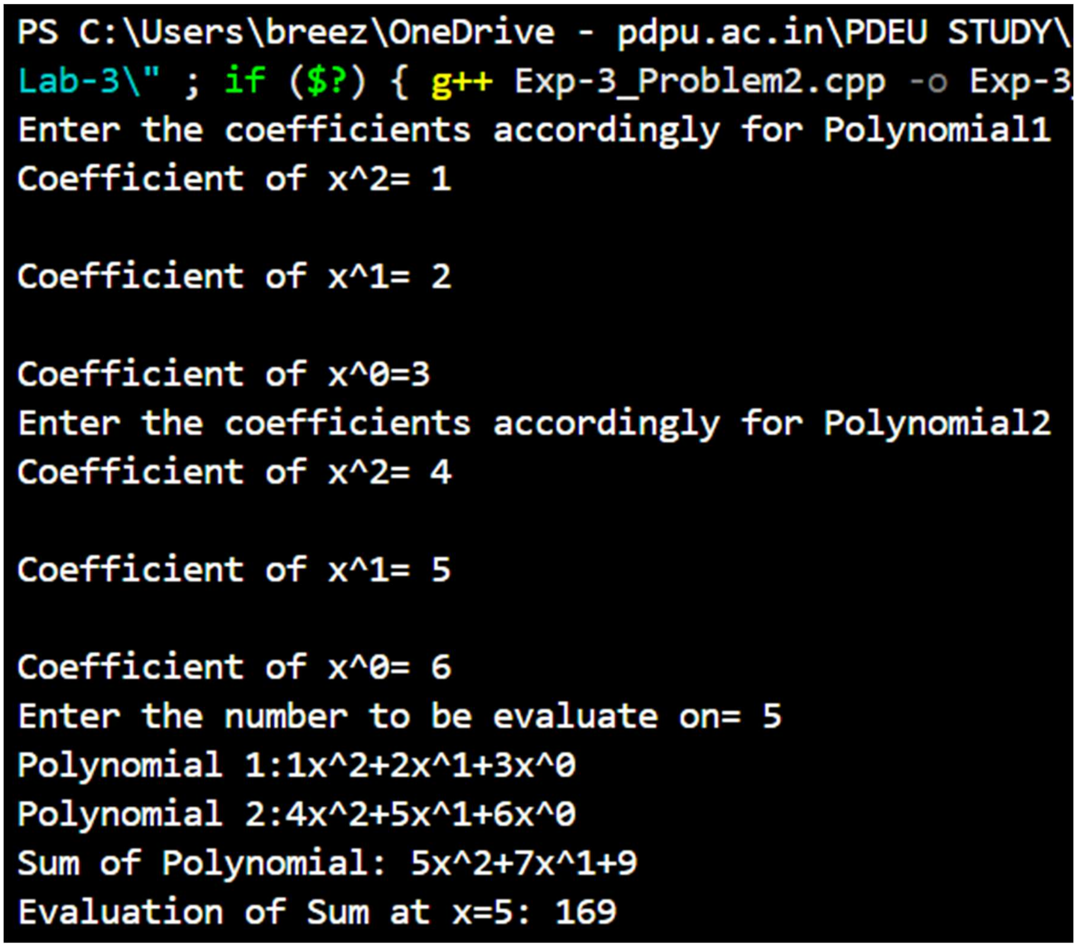
```
PS C:\Users\breez\OneDr
n\PDEU STUDY\Sem 3\OOPs
roblem1 }
Enter Name: Panav
Enter Roll Number: 128
Enter Marks: 99
Student details
Name: Panav
Roll Number: 128
Marks: 99
```

Problem 2: Implement a Polynomial class with methods to add and evaluate polynomials.

Code:

```
#include<iostream>
using namespace std;
class Polynomial
{
  int sum1=0,sum2=0,sum3=0,eval;
  public:
    void add(int p1[],int p2[])
    {
      sum1=p1[0]+p2[0];
      sum2=p1[1]+p2[1];
      sum3=p1[2]+p2[2];
      cout<<"Sum of Polynomial: "<<sum1<<"x^2+"<<sum2<<"x^1+"<<sum3;
    }

    void evaluate(int x)
    {
      eval=sum1*(x*x)+sum2*(x)+sum3;
      cout<<"\nEvaluation of Sum at x="<<x<<": "<<eval;
    }
};

int main()
{
  Polynomial p;
  int x1,y1,z1,x2,y2,z2,x;
  cout<<"Enter the coefficients accordingly for Polynomial1\n";
  cout<<"Coefficient of x^2= ";
  cin>>x1;
  cout<<"\nCoefficient of x^1= ";
  cin>>y1;
  cout<<"\nCoefficient of x^0=";
  cin>>z1;
  cout<<"Enter the coefficients accordingly for Polynomial2\n";
  cout<<"Coefficient of x^2= ";
  cin>>x2;
  cout<<"\nCoefficient of x^1= ";
  cin>>y2;
  cout<<"\nCoefficient of x^0= ";
  cin>>z2;
  cout<<"Enter the number to be evaluate on= ";
  cin>>x;
  cout<<"Polynomial 1:"<<x1<<"x^2+"<<y1<<"x^1+"<<z1<<"x^0\n";
  cout<<"Polynomial 2:"<<x2<<"x^2+"<<y2<<"x^1+"<<z2<<"x^0\n";
  int p1[]={x1,y1,z1};
  int p2[]={x2,y2,z2};
  p.add(p1,p2);
  p.evaluate(x);
  return 0;
}
```

Output:



```
PS C:\Users\breez\OneDrive - pdpu.ac.in\PDEU STUDY\
Lab-3\" ; if ($?) { g++ Exp-3_Problem2.cpp -o Exp-3
Enter the coefficients accordingly for Polynomial1
Coefficient of x^2= 1

Coefficient of x^1= 2

Coefficient of x^0=3
Enter the coefficients accordingly for Polynomial2
Coefficient of x^2= 4

Coefficient of x^1= 5

Coefficient of x^0= 6
Enter the number to be evaluate on= 5
Polynomial 1:1x^2+2x^1+3x^0
Polynomial 2:4x^2+5x^1+6x^0
Sum of Polynomial: 5x^2+7x^1+9
Evaluation of Sum at x=5: 169
```