LAB - 10

Q1. Wap to create Hierarchical Structure. Number class is base data member int X and two derived class Perfect & Strong. It will check corresponding whether no is perfect or Strong. But thing is one input will take and that single input will check both.

```
#include <iostream>
using namespace std;
class Number
{
protected:
    int x;

public:
    void getdata()
    {
        cout << "\n Enter a Number:";
        cin >> x;
    }
};
class Perfect : public Number
{
public:
    void check_perfect()
    {
        int i, div, sum = 0;
        getdata();
        for (i = 1; i < x; i++)
        {
            div = x % i;
            if (div == 0)
                sum = sum + i;
        }
        if (sum == x)
            cout << "\n " << x << " is a perfect number.";
        else
            cout << "\n " << x << " is not a perfect number.";
    }
};
class Strong : public Number
{
public:
    void check_strong()
    {
        getdata();
        int temp = x;
        int i, sum = 0;
        while (x)
        {
            int num = x % 10;
            int fact = 1;
            for (i = num; i > 0; i--)
```

```
            {
                fact = fact * i;
            }
            sum += fact;
            x /= 10;
        }
        if (sum == temp)
        {
            cout << temp << " is a Strong Number";
        }
        else
        {
            cout << temp << " is not a Strong Number";
        }
    }
};
int main()
{
    Perfect obj1;
    Strong obj;
    obj1.check_perfect();
    obj.check_strong();
    return 0;
}
```

```
Enter a Number:6

6 is a perfect number.
Enter a Number:154
154 is not a Strong Number
```

Q2. Wap to create One Base class Father data member fname. And Son class derived from Father dada member sname and GrandSon derived from Son data member gname. Use appropriate function that GrandSon will display all name by it concatenating.

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

class Fname
{

public:
    string fname1, fname2;
    Fname()
    {
        cout << "Enter Father's Name:" << endl;
        cin >> fname1;
        cin >> fname2;
    }
};

class Sname
{
```

```
public:
    string sname1;
    Sname()
    {
        cout << "Enter Son's Name:" << endl;
        cin >> sname1;
    }
};

class Gname : public Fname, public Sname
{
    string gname1;

public:
    Gname()
    {
        cout << "Enter Grandson's Name:" << endl;
        cin >> gname1;
        cout << "Grandson's Name is: " << fname1 << " " << sname1 << " " << gname1 << " " << fname2;
    }
};

int main()
{
    Gname c;
    return 0;
}
```

```
Enter Father's Name:
George
Jones
Enter Son's Name:
Trevor
Enter Grandson's Name:
Michael
Grandson's Name is: George Trevor Michael Jones
```