1)

```
#include<iostream>
using namespace std;
class Array
{
    private:
        int Arr[10];
    public:
        void getnum()
        {
            cout<<"Enter the number";
            for(int i=0;i<10;i++)
                cin>>arr[i];
        }
        void smallest()
        {
            int i, 1st ele  2nd ele;
            1st ele = 2nd ele = INT_MAX
            for(i=0;i<10;i++)
            {
                if(arr[i]<1stele)
                {
                    2ndele= 1st ele;
                    1st ele = arr[i];
                }
                else if(arr[i]<2nd_ele
                        && arr[i]!=first_ele)
                {
                    2nd ele = arr[i];
                }
            if(2ndele=INT_MIN)
            {
                cout<<"NO 2nd smallest";
            }
        }
```

```
cout << "The second
Void display()
{
  if (2ndele = int- MAX)
    { cout << "NO 2ndsmallest";
  else { cout << "The secondsmallest is " << 2nd ele;
}
int main()
{
    Array A1;
    A1. getnum();
    A1. smallest();
    A1. display();
}
```

(2)

```
#include <iostream>
using namespace std;
class Num
{
  private:
    int n;
    public:
    voidgetnum()
      { cout << "Enter the number";
        cin >> n;
    }
    Void. Length()
      { int i; count=0;
```

```
    }
  }
  if(temp==2)
  {
    cout<<i<<"\t";
  }
 }
}
3.
void Armstrong()
{
  int r, sum=0, temp;
  while(n>0)
  {
    r = n%10;
    sum = sum+(r*r*r);
    n = n/10;
  }
  if(temp==sum)
    cout<<"armstrong number";
  else
    cout<<"not armstrong";
 }
3.
int main()
{ Num N1
  N1.getnum();
  N1.Length();
  N1.palindrome();
  N1.prime();
  N1.Armstrong();
}
```

③

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

class Time
{
    private:
        int hours;
        int minutes;
        int seconds;
        char t[3];
    public:
        void getTime()
        {
            cout<<"Enter time"<<endl;
            cout<<"Hours";
            cin>>hours;
            cout<<"minutes";
            cin>>minutes;
            cout<<"seconds";
            cin>>seconds;
            cout<<"AM/PM";
            cin.ignore();
            cin.getline(t,3);
            set.Time();
        }
};
```

```
void setTime()
{
    cout<<endl;
    cout<<"Time";
    cout<<hours<<":"<<minutes<<":"
        <<seconds<<endl;
}
void addTime(TimeT1, TimeT2)
{
    seconds = T1.seconds+T2.seconds;
    minutes = T1.minutes+T2.minutes
              +(seconds/60);
    hours = T1.hours+T2.hours+(minutes/60);
    minutes = minutes%60;
    seconds = seconds%60;
    if(hours>=24)
    {
        hours = hours%24
    }
    else if(hours>=12)
    {
        hours = hours%12;
    }
}
};
int main()
{
    Time T1, T2, T3;
    T1.getTime();
    T2.getTime();
    T3.addTime(T1, T2);
    T3.setTime();
}
```

5

```
#include<iostream>
#define SIZE 10
using namespace std;
class STACK
{ private:
    int num[SIZE];
    int top;
  public:
    STACK() {
        top = -1;
    }
    int push(int n) {
        isFull()
        if (top == SIZE-1)
        {
            cout << "Full";
        }
        ++top;
        num[top] = n;
        return n;
    }
```

```
int POP()
{
    int temp;
    if(isEmpty())
    {
        cout<<"under flow";
    }
    temp= num[top];
    --top;
    return temp;
}
```

```
int isEmpty()
{
  if(top==-1)
    return 1;
  else
    return 0;
int isFull()
{
  if(top==(size-1))
    return 1;
  else return 0;
}
```

3.

```
void displayItems()
{
    int i;
    cout<<"STACK is: ";
    for(i=(top);i>=0;i--)
        cout<<num[i]<<" ";
    cout<<endl;
}
STACK()
{
    ip = new int[10];
    for(int i=0; i<10; i++)
    {
        cout<<"Enter the number";
        cin>>*(ip+i);
    }
    cout<<"numbers are";
    for(int i=0; i<10; i++)
    {
        cout<<*(ip+i);
    }
}
```

```
~STACK()
{
    for(int i=0; i<10; i++)
    {
        delete[i] ip;
    }
};
int main()
{
    STACK stk;
    int choice, n, temp;
    do
    {
        cout << endl;
        cout << "1- push" << endl;
        cout << "2- pop" << endl;
        cout << "3- Display" << endl;
        cout << "4- Exit" << endl;
        cout << "Enter your choice";
        cin >> choice;
        switch(choice)
        {
            case 1:
            cout << "Enter item to insert";
            cin >> n;
            temp = stk.push(n);
            if (temp == 0)
                cout << "STACK IS FULL" << endl;
            else
                cout << temp << "Inserted" << endl;
            break;
```

```
case 2:
    temp = stk.pop();
    if(temp==0)
        cout << "STACK IS EMPTY" << endl;
    else
        cout << temp << " is removed" << endl;
    break;
case 3:
    stk.displayItems();
    break;
case 4:
    exit(0);
    break;
default:
    cout << "An invalid choice" << endl;

}
} while(choice != 0);
return 0;
}
```

4
```
#include<iostream>
#include<cmath>
using namespace std;
class Time
{
    int hour, minute, seconds;
    public:
    void readtime()
    {
        cout<<"Enter hour min and seconds";
        cin>>hours>>minute>>seconds;
    }
    Time(int no1, int no2, int no3)
    {
        hour=no1;
        minute=no2;
        seconds=no3;
    }
    Time(){}
    friend Time operator+(Time,Time)
    void displayTime()
    {
        cout<<hour<<"hr:"<<minute<<"min:"<<seconds<<"sec";
```

```
Time operator + (Time ob1, Time ob2, Time ob3)
{
    Time ob3;
    ob3.hour = hour + ob2.hour;
    ob3.seconds = seconds + ob2.seconds;
    ob3.minutes = minutes + ob2.minutes
                  + (seconds / 60);
    ob3.hours = hours + ob1.hours +
                (minutes / 60);
    minutes = minutes % 60;
    seconds = seconds % 60;
    if(hours >= 24)
    {
        hours = hours % 24;
    }
    IF (hours >= 12)
    {
        hours = hours % 12;
    }
}
};
int main()
{
    Time ob1, ob2, ob3;
    ob1.getTime();
    ob2.getTime();
    ob3.addTime(ob1, ob2);
    ob3.setTime();
}
```

⑥

```
#Include<Iostream>
#Include<cmath>
using namespace std;

class Ref;

class Text
{
    Int NO-Book;
    float av-price;
    Public:
      void getdata1()
      {
        cout<<" Enter the NO-Book
                and av-price ";

        cin>>NO-Book;

        cin >> av-price;

      }
      void showdata1()
      {
        cout<<" NO-Book and av-price
        Entered";

        cout <<NO-Book <<av-price;

      }
      3.
      friend void add cost (Text, Ref);

};
```

```
class Ref
{
    int RefNO-Book;
    float RefAv-price;
    public:
      void getdata2()
      {
        cout << " Enter the NO-Book
                  and Av-price ";
        cin >> RefNO-Book;
        cin >> RefAv-price;
      }
      void showdata2()
      {
        cout << " NO-Book and Av-price ";
        cout << RefNO-Book << RefAv-price;
      }
      friend void addcost(Text, Ref);
};
void addcost(Text A, Ref B)
{
    float Temp = 0;
    Temp = A.av-price + B.Refav-price;
    return Temp;
}
int main()
{
    Text A;
    Ref B;
    float t;
```

```
A.getdata1();
A.showdata1();
B.getdata2();
B.showdata2();
t=addcost(A,B);
cout<<t;
}
```

```
#include<iostream>
#include<cstring>
using namespace std;
class compare string{
public:
    char str[25];
    compare string(char str1[])
    {
        int i;
        for(i=0;str1[i]!='\0';i++)
        {
            this->str[i]=str1[i];
        }
        this->str[i]='\0';
    }
```

```
int operator == (Compare string s2)
{
    If (compare( str, s2.str)==0)
        return 1;
    else
        return 0;
}
```

3.

```
char compare(str[], str[])
{
    int temp;
    for (int i=0; s1[i] != '\0' && s2[i] != '\0'; i++){
        if (s1[i] != s2[i])
            temp ++;
    }
    if(temp==0)
    {   return 1;
    }
    else
    {   return 0;
    }
}
```

3.

```
int length(char s[25])
{
    int i, count=0;
    for(i=0; ch[i]!='\0'; i++)
    {
        count++;
    }
    return count;
}
int operator<=(compare string s3)
{
    if(length(str) <= length(s3.str))
        return 1;
    else
        return 0;
}
int operator>=(compare string s3)
{
    if(length(str) >= length(s3.str))
        return 1;
    else
        return 0;
}
};
```

```
void compare( compareString s1,
                compareString s2)
{
   if (s1 == s2)
   cout << s1.str << " is equal to "
        << s2.str << endl;

   else {
   cout << s1.str << " is not equal to "
        << s2.str << endl;

   if (s1 >= s2)
   cout << s1.str << " is greater than "
        << s2.str << endl;

   else
   cout << s2.str << " is greater "
        << s1.str << endl;

}
void test case()
{
   char str1[25], str2[25];
   cout << "Enter the 1st string ";
   cin.getline(str1, 25);
   cin.ignore();
   cout << "Enter the 2nd string";
   cin.getline(str2, 25);
   cin.ignore();
```

```
compare string s1(str1);
compare string s2(str2);

cout << "comparing \"" << s1.str
     << "\" and \"" << s2.str
     << "\"" << endl;

compare (s1, s2);
}
```

```
int main()
{
    ~~input();~~
    testcase();

}
```

(8)
```
#include <iostream>
using namespace std;

class complex
{
    int real, img;
public:
    void input(int r, int img)
    {
        real = r;
        img = img;
    }
```

```
void display( )
{
    cout << "The Resultant Complex";
    cout << real << "+i" << img << endl;
}
void operator += (Complex ob2)
{
    real = ob.real;
    img = ob.img;
};
main()
{
    Complex ob1, ob2;
    cout << "Enter the real and Imaginary Part:";
    ob1.input();
    cout << "Enter the real and Imaginary Part";
    ob2.input();
    ob1.display();
    ob2.display();
    ob1 += ob2;
    ob1.display();
    Complex ob3;
    ob3 = ob2;
    ob3.display();
}
```