/*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

1. Define a class employee having the following description:

Data Members	Description
Pan	To store personal account number
Name	To store name
Taxincome	To store annual taxable income
Tax	To store tax that is calculated

Member Functions	Description
InputInfo()	Store the pan number,name,taxableincome
TaxCalc()	Calculate tax for an employee
DisplayInfo()	Output details of an employee

Write a C++ program to compute the tax according to the given conditions and display the output.

Total Annual Taxable Income	Tax Rate
Upto 2,50,000	No tax
From 2,50,000 to 3,00,000	10 % of the income exceeding 2,50,000
From 3,00,000 to 4,00,000	Rs. 5000+20 % of the income exceeding 3,00,000
Above 4,00,000	Rs 25000 + 30 % of the income exceeding 4,00,000

*/

```
#include<iostream>
using namespace std;
class employee
{
   int taxincome;
   float tax;
   string pan, name;
   public:
      void inputinfo()
      {
         cout<<"Enter personal account number: ";
         cin>>pan;
         cout<<"Enter store name: ";
         cin>>name;
         cout<<"Enter annual taxable income: ";
         cin>>taxincome;
      }
      void taxcalc()
```

```
    {
        if (taxincome<=250000)
            tax=0;
        else if (taxincome>250000 && taxincome<=300000)
            tax=0.1*(taxincome-250000);
        else if (taxincome>300000 && taxincome<=400000)
            tax=5000+(0.2*(taxincome-300000));
        else if (taxincome>400000)
            tax=25000+(0.3*(taxincome-400000));
    }
    void displayinfo()
    {
        cout<<"--------------------------------------\n";
        cout<<"Personal Account Number: "<<pan<<"\n";
        cout<<"Store Name: "<<name<<"\n";
        cout<<"Annual Taxable Income: "<<taxincome<<"\n";
        cout<<"Tax: "<<tax<<"\n";
        cout<<"--------------------------------------\n";
    }
};
int main()
{
    employee e;
    e.inputinfo();
    e.taxcalc();
    e.displayinfo();
    return 0;
}
```

*****OUTPUT*****

Enter personal account number: ABC123

Enter store name: Rebook

Enter annual taxable income: 500000

Personal Account Number: ABC123

Store Name: Rebook

Annual Taxable Income: 500000

Tax: 55000

/*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

2. Define a class named FriendDemo with following description:
   Data Members Description
   string str Member Functions Description
   void setstr(string) This function will assign a sentence into str. Friend Class Description friend class VowelCheck
   Define another class named VowelCheck with following description:
   Member Functions Description
   void maxVowel(FriendDemo) Display the word of the sentence that contains maximum number of vowels. Sample Input: HAPPY NEW YEAR Sample Output: The word with maximum number of vowels: YEAR

*/

```
#include<iostream>
using namespace std;
class FriendDemo
{
   string str;
   public:
      void setstr(string s)
      {
         str=s;
      }
      friend class VowelCheck;
};
class VowelCheck
{
   int i, j=0, c=0, max=c;
   char cpy[100];
   string maxstr;
   public:
      void maxVowel(FriendDemo &f)
      {
         for(i=0; f.str[i]!='\0'; i++)
         {
```

```
            if(f.str[i]=='A' || f.str[i]=='E' || f.str[i]=='I' || f.str[i]=='O' || f.str[i]=='U')
                c++;
            if(f.str[i]==' ')
            {
                if(max<c)
                {
                    max=c;
                    maxstr=cpy;
                }
                c=0, j=0;
            }
            else
            {
                cpy[j++]=f.str[i];
                cpy[j]='\0';
            }
        }
        cout<<"The word with maximum number of vowels: "<<maxstr;
    }
};
int main()
{
    string str;
    cout<<"Enter string: ";
    getline(cin, str);
    FriendDemo f;
    f.setstr(str);
    VowelCheck v;
    v.maxVowel(f);
    return 0;
}
```

*****OUTPUT*****

```
Enter string: HAPPY NEW YEAR
The word with maximum number of vowels: YEAR
```

/*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

3. Create a class Complex having two int type variable named real & img denoting real and imaginary part respectively of a complex number. Overload + , - , = = operator to add, to subtract and to compare two complex numbers being denoted by the two complex type objects.

*/

```
#include<iostream>
using namespace std;
class complex
{
   int real, img;
   public:
      complex()
      {
         real=0, img=0;
      }
      complex(int r, int i)
      {
         real=r, img=i;
      }
      complex operator+(complex C1)
      {
         complex C3;
         C3.real=C1.real+real;
         C3.img=C1.img+img;
         return C3;
      }
      complex operator-(complex C1)
      {
         complex C4;
         C4.real=real-C1.real;
         C4.img=img-C1.img;
         return C4;
```

```
        }
        void operator==(complex C1)
        {
            if(C1.real==real && C1.img==img)
                cout<<"True";
            else
                cout<<"False";
        }
        void show()
        {
            cout<<"Result = "<<real<<"+j"<<img<<"\n";
        }
};
int main()
{
    int i, x, y;
    complex C1(40, 15), C2(20, 10);
    complex C3=C1+C2;
    C3.show();
    complex C4=C1-C2;
    C4.show();
    C2.operator==(C1);
    return 0;
}
```

*****OUTPUT*****

```
Result = 60+j25
Result = 20+j5
False
```