

Exo1

Code source

```
#include <iostream>

using namespace std;

class Complexe {

public :
    float img;
    float rel;

    Complexe(){};
    Complexe(float rel ,float img ){
        this->img=img;
        this->rel=rel;
    };

    Complexe operator-(Complexe C){
        Complexe S;
        S.img=img-C.img;
        S.rel=rel-C.rel;

        return S;}

    Complexe operator/(Complexe C){
        Complexe D;
        D.img=img/C.img;
        D.rel=rel/C.rel;
        return D;
    }

    Complexe operator*(Complexe C){

        Complexe M;
        M.img=img*C.img;
        M.rel=rel*C.rel;
        return M;}
```

```

Complexe operator+(Complexe C){

    Complexe A;
    A.img=img+C.img;
    A.rel=rel+C.rel;
    return A;}};
int main()
{
    float x1,y1;
    float x2,y2;

    cout<<"entrer la valeur du partie reel du premier nombre complex = ";
    cin>>x1;
    cout<<"entrer la valeur du partie imaginaire du premier nombre complex = ";
    cin>>y1;

    cout<<"entrer la valeur du partie reel du deuxieme nombre complex =";
    cin>>x2;
    cout<<"entrer la valeur du partie  imaginaire du deuxieme nombre complex =";
    cin>>y2;

    Complexe c1(x1,y1);
    cout<<"le premier nombre complex est :
"<<c1.rel<<"+"<<c1.img<<"i"<<"\n"<<endl;
    Complexe c2(x2,y2);
    cout<<"le deuxieme nombre complex est
:"<<c2.rel<<"+"<<c2.img<<"i"<<"\n"<<endl;

    Complexe c3;

    c3=c1+c2;

    cout<<"l'addition des deux nombre complex est
:"<<c3.rel<<"+"<<c3.img<<"i"<<"\n"<<endl;
    c3=c1-c2;

    cout<<"la soustraction des deux nombre complex est
:"<<c3.rel<<"+"<<c3.img<<"i"<<"\n"<<endl;

    c3=c1*c2;

```

```

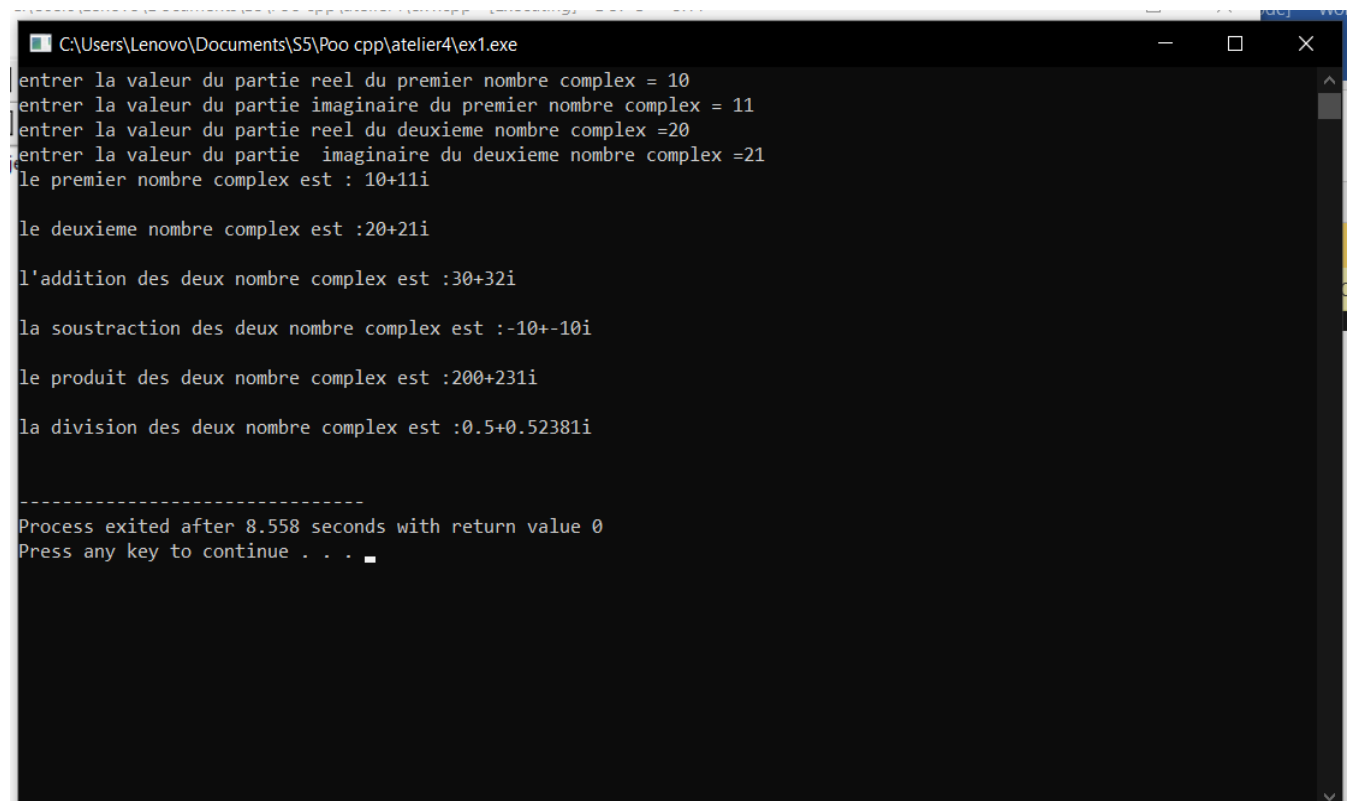
    cout<<"le produit des deux nombre complex est
:"<<c3.re<<"+"<<c3.img<<"i"<<"\n"<<endl;

    c3=c1/c2;

    cout<<"la division des deux nombre complex est
:"<<c3.re<<"+"<<c3.img<<"i"<<"\n"<<endl;
    return 0;
}

```

## Execution



```

C:\Users\Lenovo\Documents\S5\Poo cpp\atelier4\ex1.exe
entree la valeur du partie reel du premier nombre complex = 10
entree la valeur du partie imaginaire du premier nombre complex = 11
entree la valeur du partie reel du deuxieme nombre complex =20
entree la valeur du partie imaginaire du deuxieme nombre complex =21
le premier nombre complex est : 10+11i

le deuxieme nombre complex est :20+21i

l'addition des deux nombre complex est :30+32i

la soustraction des deux nombre complex est :-10+-10i

le produit des deux nombre complex est :200+231i

la division des deux nombre complex est :0.5+0.52381i

-----
Process exited after 8.558 seconds with return value 0
Press any key to continue . . .

```

## Exo2

### Code source

```
#include <iostream>
#include <string>

using namespace std;

void afficherDateEtHeure(string s)
{
    if ( s.length() != 12 )
        cerr << "Il faut entrer 12 caractere" << endl; //afficher sil ya un erreur
    else
    {
        cout << "Date : "
              << s.substr(0,2) //le jour commence par le chiffre de pos 0 et contient 2
caracteres dans sa longueur
              << "/"
              << s.substr(2,2) //le mois commence par le chiffre de pos 2 et contient
2 caracteres dans sa longueur
              << "/"
              << s.substr(4,4) << endl; //annee commence par le chiffre de pos 4 et
contient 4 caracteres dans sa longueur
        cout << "Heure : "
              << s.substr(8,2) // heure commence par le chiffre de pos 8 et contient 2
caracteres dans sa longueur
              << ":" << s.substr(10,2) << endl; // min commence par le chiffre de
pos 1 et contient 2 caracteres dans sa longueur
    }
}

int main(){
    string c;

    cout<<"Entrer un chaine"<<endl;
    cin>>c;

    afficherDateEtHeure(c);
}
```

## Execution

```
Entrer un chaine
101120211530
Date  : 10/11/2021
Heure : 15h30
```

## Exo3

### Code source

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

void triabull(int N,int a[]){
    int i;

    int tmp,nb;
    do {
        nb=0;
        for (i=0; i<N-1; i++) {
            if (a[i] > a[i+1]) {
                tmp = a[i];
                a[i] = a[i+1];
                a[i+1] = tmp;
                nb++;
            }
        }
    } while (nb!=0);

    cout << "Voici le tableau trié : " << endl;
    for (i=0; i<N; i++) {
        cout << "a[" << i << "]= " << a[i] << endl;
    }
}

void triinsertion(int N,int a[]){
    int i,j;
    for(i=0; i<N; i++) {
```

```

        cout << "Veuillez taper l'entier numero " << i << " : ";
        cin >> a[i];
    }
    int tmp;
    for (i=1 ; i <= N-1; i++) {
        j = i;

        while (j > 0 && a[j-1] > a[j]) {
            tmp = a[j];
            a[j] = a[j-1];
            a[j-1] = tmp;

            j--;
        }
    }
    for (i=0; i<N; i++) {
        cout << "a[" << i <<"]=" << a[i] << endl;
    }
}

void triselection(int N,int a[]){
    int i,min,tmp;
    for(i=0; i<N; i++) {
        cout << "Veuillez taper l'entier numero " << i << " : ";
        cin >> a[i];
    }
    for(int i = 0 ; i < N-1 ; i++)
    {
        min = i;
        for(int j = i+1 ; j < N ; j++)
            if(a[j] < a[min])
                min = j;
        if(min!=i)
        {
            //◆changer t[i] et t[min]
            tmp = a[i];
            a[i]=a[min];
            a[min]=tmp;
        }
    }
}

void menu(){
    cout<<"1.tri a bull"<<endl;
    cout<<"2.tri a insertion"<<endl;
    cout<<"3.tri de selection"<<endl;
}

```

```

int main()
{
    int N,c,a[30],i;

    cout<<"entrer le nombre des entiers a trier";
    cin>>N;
    for(i=0; i<N; i++) {
        cout << "Veuillez taper l'entier numero " << i << " : ";
        cin >> a[i];
    }
    menu();
    cout<<"entrer votre choix"<<endl;
    cin>>c;
    switch(c)
    {
        case 1:
            cout<<"-----tri a bulles -----"<<endl;
            triabull(N,a);
            break;
        case 2:
            cout<<"-----tri a insertion -----"<<endl;
            triinsertion(N,a);
            break;
        case 3:
            cout<<"-----tri a selection -----"<<endl;
            triselection(N,a);
            break;
        default :
            printf("\n Erreur verifier votre choix \n");
            break;
    }

    return 0;
}

```

Execution

```
entrer le nombre des entiers a trier3
Veuillez taper l'entier numero 0 : 5
Veuillez taper l'entier numero 1 : 3
Veuillez taper l'entier numero 2 : 8
1.tri a bull
2.tri a insertion
3.tri de selection
entrer votre choix
1
-----tri a bulles -----
Voici le tableau trié :
a[0]=3
a[1]=5
a[2]=8
-----
```

## Exo4

Code source

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

class Personne {
public :
    string nom;
    string prenom;
    int age ;
    Personne(string nom ,string prenom, int age ){
        this->nom = nom ;
        this->prenom =prenom ;
        this->age = age;
    }
};

int main()
{
```



```

string nom;
string prenom;
int age ;
int nbr;
list<Personne> data;
cout<<"entrer le nombre de personne a stocker"<<endl;
cin>>nbr;
for (int i = 0; i < nbr; ++i)
{
    cout<<"entrer prenom "<<i+1<<endl;
    cin>>prenom;
    cout<<"entrer nom "<<i+1<<endl;
    cin>>nom;
    cout<<"entrer age "<<i+1<<endl;
    cin>>age;

    data.push_back(Personne(prenom , nom,age ));
}
list <Personne> :: iterator it;
for (it = data.begin(); it != data.end(); ++it){

    cout << it->nom<<" ";
    cout << it->prenom<<" ";
    cout << it->age<<"ans"<<endl;
}
return 0;
}

```

## Execution

```

entrer le nombre de personne a stocker
1
entrer prenom 1
mohamed
entrer nom 1
ahmai
entrer age 1
15
mohamed ahmai 15ans
-----

```

## Exo5

### Code source

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

bool rechercher(int a,set<int> my_set){
if(my_set.find(a)==my_set.end()){
    return false;
}
else {
    return true;
}
}

int main(){
set<int>set_nombre;
for( int i=0 ; i<101;i++ ) {
set_nombre.insert(i);
}

for (auto it = *set_nombre.begin(); it != *set_nombre.end(); it++ ) {
    cout<<it<<"-";
}

    return 0;
}
```

## Exo6

### Code source

```
#include <iostream>
using namespace std;
class Test{
    public:
        static int tableau[] ;
    public :
```

```

        static int division(int indice, int diviseur)
        {
            try{
                if (diviseur==0)
                    throw "erreur";
                return tableau[indice]/diviseur;
            }
            catch(...){
                cout<<"ERROR: attempt to divide by zero";
            }
        }
    };
    int Test::tableau[] = {17, 12, 15, 38, 29, 157, 89, -22, 0, 5} ;
    int main() {
        int x, y;
        cout << "Entrez l indice de l entier a diviser: " << endl;
        cin >> x ;
        cout << "Entrez le diviseur: " << endl;
        cin >> y ;
        cout << "Le resultat de la division est: "<< endl;
        cout <<Test::division(x,y) << endl;

        return 0; }

```

## Execution

```

Entrez l indice de l entier a diviser:
4
Entrez le diviseur:
0
Le resultat de la division est:
ERROR: attempt to divide by zero1

```

## Exo7

### Code source

```

#include<iostream>
using namespace std;

```

```

class Point{
    private:
        double x,y;
    protected:
        string nom;
    public:
        Point(double x=0,double y=0,string nom='') {
            this->x=x;
            this->y=y;
            this->nom=nom;
        }
        double getX(){
            return x;
        }
        void setX(double x){
            this->x=x;
        }
        double getY(){
            return y;
        }
        void setY(double y){
            this->y=y;
        }
        string getNom(){
            return nom;
        }
        void setNom(string nom){
            this->nom=nom;
        }
        void Afficher(){
            cout<<"x= "<<x<<" y= "<<y<<" nom: "<<nom;
        }
};

class PointCouleur:public Point{
    private:
        unsigned int couleur;
    public:
        PointCouleur(double x=0,double y=0,string nom='',unsigned int
couleur=0):Point(x,y,nom){
            this->couleur=couleur;
        }
        unsigned int getcouleur(){
            return couleur;
        }
}

```

```
void setcouleur(unsigned int couleur){  
    this->couleur=couleur;  
}  
void Afficher(){  
    Point :: Afficher;  
    cout<<"couleur= "<<couleur;  
}  
};
```