

ATELIER 4

EXERCICE 1 :

```
1  #include <iostream>
2
3  using namespace std;
4
5  // definition de class
6  class Complex
7  {
8  public:
9      float reel;
10     float imaginaire;
11     void setValue(float re,float im);
12     Complex operator+(Complex);
13     Complex operator-(Complex);
14     Complex operator*(Complex);
15     Complex operator/(Complex);
16 };
17
18 // fonction pur changer les variable reel,imaginaire
19 void Complex::setValue(float re,float im)
20 {
21     reel = re;
22     imaginaire = im;
23 }
24
25 // Definition de l'opérateur
26 Complex Complex::operator+(Complex second){
27     Complex result;
28     result.imaginaire = this->imaginaire + second.imaginaire;
29     result.reel = this->reel + second.reel;
30     return result;
31 }
32
33 Complex Complex::operator-(Complex second){
34     Complex result;
35     result.imaginaire = this->imaginaire - second.imaginaire;
36     result.reel = this->reel - second.reel;
37     return result;
38 }
39
40 Complex Complex::operator*(Complex second){
41     Complex result;
42     result.reel = this->reel * second.reel - this->imaginaire * second.imaginaire;
43     result.imaginaire = this->reel * second.imaginaire + this->imaginaire * second.reel;
44     return result;
45 }
```

```

} Complex Complex::operator/(Complex second){
    Complex result;
    result.reel = (this->reel * second.reel + this->imaginaire * second.imaginaire) /
        (second.reel * second.reel + second.imaginaire * second.imaginaire);
    result.imaginaire = (this->imaginaire * second.reel - this->reel * second.imaginaire) /
        (second.reel * second.reel + second.imaginaire * second.imaginaire);
    return result;
}

//fonction prend deux parametre de type (class) Complex
// et retourne un variable de meme type Complex
Complex arithmetics(Complex first,Complex second){
    Complex result; // pour stocker la resultat
    float tempRe,tempIm;
    char operation ;
    //on montre les operations possible a l'utilisateur
    cout << "quelle est votre operation ? \n";
    cout << "1: la somme\n";
    cout << "2: la soustraction\n";
    cout << "3: la multiplication\n";
    cout << "4: la devision\n";
    cin >> operation;

    // on applique directement les regle des operation des nombres complex
    // selon l'operation choisi par l'utilisateur
    switch (operation)
    {
        case '1':
            result = first + second;
            break;
        case '2':
            result = first - second;
            break;
        case '3':
            result = first * second;
            break;
        case '4':
            result = first / second;
            break;
    }

    return result;
}

```

```

int main(int argc, char const *argv[])
{
    float tempRe,tempIm;

    // creation de deux nombre complex
    Complex z1,z2;

    // on demande des valeur
    cout << "entre la partie reel du premier nombre : " << endl;
    cin >> tempRe;
    cout << "entre la partie imaginaire du premier nombre : " << endl;
    cin >> tempIm;
    z1.setValue(tempRe,tempIm);

    cout << "entre la partie reel du deuxieme nombre : " << endl;
    cin >> tempRe;
    cout << "entre la partie imaginaire du deuxieme nombre : " << endl;
    cin >> tempIm;
    z2.setValue(tempRe,tempIm);

    // appele de la fonction
    cout << "le premier nombre: " << z1.reel << " + " << z1.imaginaire << "i \n";
    cout << "le deuxieme nombre: " << z2.reel << " + " << z2.imaginaire << "i \n";
    Complex result = arithmetics(z1,z2);

    // affichage de la resultat de l'operation
    cout << "Votre solution est : " << endl << result.reel << " + " << result.imaginaire << "i \n";

    return 0;
}

```

```

entre la partie reel du premier nombre :
2
entre la partie imaginaire du premier nombre :
13
entre la partie reel du deuxieme nombre :
4
entre la partie imaginaire du deuxieme nombre :
18
le premier nombre: 2 + 13i
le deuxieme nombre: 4 + 18i
quelle est votre operation ?
1: la somme
2: la soustraction
3: la multiplication
4: la devision

```

EXERCICE 2 :

```

1  #include <iostream>
2  #include <iomanip>
3  #include <string>
4
5  using namespace std;
6
7  void afficherDateEtHeure(const string& s)
8  {
9      if ( s.length() != 12 )
10         cerr << "Chaine invalide." << endl;
11     else
12     {
13         cout << "Date : " << s.substr(0,2) << "/" << s.substr(2,2) << "/" << s.substr(4,4) << endl;
14         cout << "Heure : " << s.substr(8,2) << "h" << s.substr(10,2) << endl;
15     }
16 }
17
18 int main(int argc, char** argv)
19 {
20     string s ;
21     cout << "Entrer la forme JJMMAAAHHNN : " << endl ;
22     cin >> s;
23     afficherDateEtHeure(s); // exemple
24 }
25

```

C:\Users\pc\Desktop\s5\c++\ATELIER-4\PROG-2.exe

```

Entrer la forme JJMMAAAHHNN :
120219990512
Date : 12/02/1999
Heure : 05h12

-----
Process exited after 7.032 seconds with return value 0
Appuyez sur une touche pour continuer...

```

EXERCICE 4 :

```

1  #include <iostream>
2  #include <list>
3  using namespace std;
4  int main()
5  {
6      list<string> Liste; //d  clarer une liste
7      string n;
8      int i;
9      cout<<"Entrer N, Prenom et Age de chaque personne sous form de Nom,Prenom,age. inserez ( pause ) quand vous terminez : "<<endl;
10     for ( i = 0; i++ )
11     {
12         cin>>n;
13         if(n=="pause")
14         {
15             break;
16         }
17         Liste.push_back(n);
18     }
19     list<string>::iterator it;
20     Liste.sort();
21     it=Liste.begin();
22     cout<<"votre Triage est : " <<endl ;
23     for (it;it!=Liste.end();it++)
24     {
25         cout<<*it<<" ";
26     }
27 }

```

```
Entrer N, Prenom et Age de chaque personne sous form de Nom,Prenom,age. inserez ( pause ) quand vous terminez :
CHARKAOUI,WIAM,22
PAUSE
pause
votre Triage est :
CHARKAOUI,WIAM,22 PAUSE
-----
Process exited after 30.72 seconds with return value 0
Appuyez sur une touche pour continuer... █
```

EXERCICE 6 :

```
1  #include <iostream>
2  #include <stdexcept>
3
4  using namespace std;
5
6  class Test{
7  public:
8      static int tableau[] ;
9  public :
10     static int division(int indice, int diviseur){
11         if (diviseur==0)
12         {
13             throw invalid_argument("Erreur, division par 0!") ;
14         }
15         return tableau[indice]/diviseur;
16     }
17 };
18
19
20 int Test::tableau[] = {17, 12, 15, 38, 29, 157, 89, -22, 0, 5} ;
21
22 int main()
23 {
24     int x, y;
25     cout << "Entrez l'indice de l'entier a diviser: " << endl;
26     cin >> x ;
27     cout << "Entrez le diviseur: " << endl;
28     cin >> y ;
29
30     try
31     {
32         int temp = Test::division(x,y);
33         cout << "Le resultat de la division est: "<< endl;
34         cout << temp << endl;
35     }
36     catch(invalid_argument u)
37     {
38         cerr << u.what() << '\n';
39     }
40
41     return 0;
42 }
```

C:\Users\pc\Desktop\s5\c++\ATELIER-4\PROG-6.exe

Entrez l'indice de l'entier a diviser:

15

Entrez le diviseur:

2

Le resultat de la division est:

537

Process exited after 9.131 seconds with return value 0

Appuyez sur une touche pour continuer... ■