

TP n° 16 – Structures

Exercice 1 : surcharge de fonction

Soit la fonction **main** suivante :

```
int main(void)
{
int i1,i2,i3,somI;
cout<<"veuillez saisir 3 entiers"<<endl;
cin>>i1>>i2>>i3;
somI=somme(i1,i2);
cout<<i1<<" + "<<i2<<" = "<<somI<<endl;
somI=somme(i1,i2,i3);
cout<<i1<<" + "<<i2<<" + "<<i3<<" = "<<somI<<endl;

float f1,f2,f3,somF;
cout<<"veuillez saisir 3 réels"<<endl;
cin>>f1>>f2>>f3;
somF=somme(f1,f2);
cout<<f1<<" + "<<f2<<" = "<<somF<<endl;
somF=somme(f1,f2,f3);
cout<<f1<<" + "<<f2<<" + "<<f3<<" = "<<somF<<endl;

long l1,l2,l3,somL;
cout<<"veuillez saisir 3 longs entiers"<<endl;
cin>>l1>>l2>>l3;
somL=somme(l1,l2);
cout<<l1<<" + "<<l2<<" = "<<somL<<endl;
somL=somme(l1,l2,l3);
cout<<l1<<" + "<<l2<<" + "<<l3<<" = "<<somL<<endl;

return 0 ;
}
```

Écrire les fonctions **somme** correspondantes.

Exercice 2 : conversion de type

Écrire et tester le programme ci-dessous :

```
1:  #include<iostream>
2:  #include<string>
3:  using namespace std;
4:
5:  int main(void)
6:  {
7:  char a='a';
8:  cout << a << endl;
9:  char A='a'-32;
10: cout << A << endl;
11: for (int i=128; i<255; i++)
12: {
13:     cout << (char)i << " " << i << endl;
14: }
15: return 0 ;
16: }
```

A quoi sert l'instruction `(char)` avant le `i` à la ligne 13?

Exercice 3 : structure

Écrire les types structurés suivants :

- Rectangle : contenant les champs longueur, largeur, perimetre et aire
- Cercle : contenant les champs rayon, perimetre et aire.

Écrire les fonctions `CalculePerimetre` et `CalculeAire` permettant de calculer les périmètres et aires des rectangles et des cercles.

Exercice 4 : Notes

Écrire un programme facilitant la saisie des notes dans un établissement. Pour cela vous définirez une structure `matiere` contenant les champs :

- `nom` : contenant le nom de la matière
- `prof` : contenant le nom du professeur
- `elevés` : contenant les noms des élèves (utilisez un tableau de string...)
- `notes` : contenant les notes de ces élèves dans cette matière (on suppose qu'il y a 3 notes par élèves et qu'il y a 5 élèves).

Vous écrirez également les fonctions suivantes :

- `initialisation` : ne prenant aucun paramètre, la fonction doit permettre d'initialiser une matière en demandant à l'utilisateur de saisir le nom de la matière, le nom du professeur et le nom de chaque élève. A la fin de cette initialisation, la fonction renvoie la variable contenant toutes ces informations.
- `CreerMotDePasse` : prenant en paramètre une variable de type `matiere` et fourni un mot de passe à l'utilisateur correspondant à la deuxième et troisième lettre de la matière, suivi des trois dernières lettres du nom du professeur.
- `validiteMotDePasse` : prenant en paramètre une variable de type `matiere` et une chaîne de caractère et renvoyant vrai ou faux selon si la chaîne de caractère correspond au mot de passe associé à la variable de type `matiere` ou non.
- `saisirNotes` : prenant en paramètre une variable de type `matiere` et une chaîne de caractère, demandant à l'utilisateur de saisir un mot de passe et s'il est correcte lui permettant de saisir les notes de l'élève ayant pour nom la chaîne de caractère.
- `afficherNotes` : prenant en paramètre passé une variable de type `matiere`, demandant à l'utilisateur de saisir un mot de passe et s'il est correcte affichant les notes de tous les élèves.
- `afficherMoyenne` : prenant en paramètre passé une variable de type `matiere`, demandant à l'utilisateur de saisir un mot de passe et s'il est correcte affichant la moyenne dans cette matière.

Enfin, vous écrirez une fonction principale permettant à l'utilisateur, à l'aide d'un menu, d'effectuer les différentes actions proposées.