

# Contrôle individuel : Adaptation langage C++

24 Octobre 2012

1) Considérer un tableau de notes de type virgule flottante. On utilise le chiffre -1 pour signaler la fin des notes :

```
float notes[TAB_MAX] = { 15,10,2,2.5,18,5,11.5,15,18,8,4,-1 } ;
```

- a) Ecrire une fonction « affiche » qui affiche à l'écran les valeurs du tableau et qui retourne le nombre de notes.  
**int** affiche (**float** tab[]){ ... }
- b) Ecrire une nouvelle fonction, « saisie », qui demande à l'utilisateur d'entrer les notes. On se sert d'une entrée de -1 afin d'arrêter la saisie des notes.  
**void** saisi (**float** ntab[]){ ... }
- c) Ecrire une fonction « moyenne » qui prend le tableau de notes comme argument et qui retourne la moyenne des notes (de type **float**).  
**float** moyenne (**float** tab[]){ ... }
- d) Modifier la fonction « moyenne » afin qu'elle retourne les notes maximales et minimales (passage par référence des paramètres).  
**float** moyenne(**float** tab[],**float**& min, **float**& max) ;

```
const int TAB_MAX=20 ;
```

```
main()
```

```
{  
    int nbre;  
    float notes[TAB_MAX] = { 15,10,2,2.5,18,5,11.5,15,18,8,4,-1 } ;  
    float moy, minnote,maxnote ;  
    ⋮  
    saisie(notes);  
    nbre = affiche (notes) ;  
    moy = moyenne( notes, minnote, maxnote) ;  
    cout << "La moyenne des " << nbre << " notes = " << moy << endl ;  
    ⋮  
}
```

2) Ecrire un programme qui demande l'utilisateur d'entrer un entier  $n$  qui servira comme argument d'une fonction « harm\_somme » qui prendra  $n$  pour argument et qui retournera la valeur de la somme harmonique :

$$\sum_{k=1}^n \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

Suivre le modèle suivant :

⋮

```
double harm_somme (int nmax){ ... }
```

```
main()
```

```
{
```

```
    double somme ;
```

```
    int n ;
```

```
    ⋮
```

```
    somme = harm_somme( n ) ;
```

```
    ⋮
```

```
}
```

a) Donner les résultats trouvés par le programme pour  $n= 10, 100, 1000, 1000000$ .  
(Attn ! : en langage C/C++  $1/2=1/3=\dots 1/n =0, \dots$  mais  $1./2=0.5, 1./3 = 0.333333,\dots$  )

b) Ajouter une fonction « log\_somme » qui calcule la somme de la suite:

$$\sum_{k=1}^n \frac{(-1)^{(k+1)}}{k} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{(-1)^{(n+1)}}{n}$$

Donner les résultats de la suite pour  $n= 10, 100, 1000$ . Comparer ces valeurs à  $\log(2)$ .

3) Considérer la chaîne de caractères suivante qui sera déclarée dans la fonction main :

```
char phrase[]="You will hopefully feel better when this exam is over :). *** This "
```

```
"character chain is in English because the C++ programming language uses"
```

```
" ASCII coding which doesn't handle French accents like é, à, and ç, and ô very well";
```

(On se rappelle que la fin de n'importe quelle chaîne de caractères en langage C++ est terminée de façon automatique par le caractère '\0' afin qu'on puisse détecter la fin de la chaîne.)

a) Ecrire une fonction, « nombrechar », qui retourne à la fonction main, le nombre total de caractères alphabétiques, incluant les minuscules ('a' 'b', 'c' , ... , 'z') et les majuscules ('A', 'B' , ... , 'Z').

```
int nombrechar (char chaine[]){ ... }
```

b) Ecrire une fonction, « comptechar », qui retourne le nombre de fois un caractère donné apparaît dans la chaîne de caractères. (Essaie avec  $ch = 'e'$ )

```
int comptechar (char chaine[],char ch){ ... }
```