

TD3 – Tests et Boucles

Partie TD

Exercice 1 – Mentions (tests sur intervalle de valeurs)

Exercice 2 – Affichage de sinus et de cosinus d'angles

Exercice 3 – Lecture contrôlée d'un nombre

Exercice 4 – Calcul de $\sum_{i=a}^b i = a + (a + 1) + \dots + (b - 1) + b$

Exercice 5 – Calcul de $\prod_{i=a}^b i = a \times (a + 1) \times \dots \times (b - 1) \times b$

Exercice 6 – Calcul de n^m

Partie TME

Exercice 7 – Nombre d'occurrences d'un entier dans une saisie

Exercice 8 – Calculatrice simple

Exercice 9 – Jeu : devinette

Annexes : corrections

Correction de l'exercice [mention]

Correction de l'exercice [cos]

Correction de l'exercice [Devinette]

Partie TD

Exercice 1 – Mentions (tests sur intervalle de valeurs)

Ecrire un programme affichant la mention (non admis, passable, assez bien, bien...) en fonction d'une note x sur 20 (demi-points acceptés) entrée par l'utilisateur.

Une solution est fournie en [annexe](#).

Exercice 2 – Affichage de sinus et de cosinus d'angles

Ecrire un programme qui affiche les sinus et les cosinus des angles de 0 à 90 degrés par pas de 10 degrés.

Une correction est donnée en [annexe](#).

Exercice 3 – Lecture contrôlée d'un nombre

Ecrire un programme demandant à l'utilisateur d'entrer un nombre tant que ce nombre est négatif ou supérieur à 20.

Exercice 4 – Calcul de $\sum_{i=a}^b i = a + (a + 1) + \dots + (b - 1) + b$

Ecrire deux programmes (en utilisant une boucle "Pour" puis une boucle "Tant que") calculant la somme des entiers compris entre deux nombres a et b entrés par l'utilisateur. Que se passe-t-il si $a > b$?

Exercice 5 – Calcul de $\prod_{i=a}^b i = a \times (a + 1) \times \dots \times (b - 1) \times b$

Ecrire deux programmes (en utilisant une boucle "Pour" puis une boucle "Tant que") calculant le produit des entiers compris entre deux nombres a et b rentrés par l'utilisateur. Que se passe-t-il si $a > b$?

Exercice 6 – Calcul de n^m

Ecrire un programme permettant de calculer n^m (n et m entiers positifs ou nuls) sans utiliser la fonction `pow(,)` fournie par la bibliothèque `math.h`.

Partie TME

Exercice 7 – Nombre d’occurrences d’un entier dans une saisie

Un opérateur saisit des entiers non nuls au clavier. Il indique que la saisie des entiers est terminée en entrant \emptyset . Proposer un programme permettant de chercher le nombre d’occurrences d’un entier i (connu à l’avance, et différent de \emptyset , par exemple $i = 3$) dans la série d’entiers saisie par l’opérateur.

Indication : Le nombre d’occurrences correspond au nombre d’apparitions du nombre i . Pour traiter cet exercice, vous devez trouver une stratégie qui permet de ne pas avoir à stocker en mémoire toute la série d’entiers.

On ne connaît pas le nombre d’itérations et il faut faire au moins un tour de boucle, donc boucle “Répéter ... jusqu’à ce que”.

Exercice 8 – Calculatrice simple

Ecrire un programme en C qui calcule et affiche le résultat d’une opération élémentaire entre deux entiers. On demandera à l’utilisateur de saisir les deux entiers a et b , ainsi que l’opérateur arithmétique (sous forme de caractère $+$, $-$, $*$ ou $/$).

Exercice 9 – Jeu : devinette

Ecrire un programme qui demande à l'utilisateur de choisir mentalement un entier entre 0 et 1000, puis qui détermine ce nombre en posant des questions à l'utilisateur et en se rapprochant de la solution par dichotomie, suivant le programme (en pseudo-code) ci-dessous.

Modèle de programme 1 Devinette

```

1: Demander à l'utilisateur de retenir un entier entre 0 et 1000
2:  $g \leftarrow 0$  // indice gauche
3:  $d \leftarrow 1000$  // indice droite
4: TANT QUE  $g \leq d$  FAIRE
5:    $m \leftarrow (g + d)/2$  //indice milieu
6:   SI  $m == \text{nombre}$  ALORS
7:     nombre trouve
8:   SINON
9:     SI  $\text{nombre} < m$  ALORS
10:       $d \leftarrow m - 1$ 
11:     SINON
12:       $g \leftarrow m + 1$ 
13:     FIN SI
14:   FIN SI
15: FIN TANT QUE

```

Une solution est donnée en [annexe](#).

Annexes : corrections

Correction de l'exercice [[mention](#)]

```

#include <stdio.h>
// Determination de la mention en fonction de la note
int main () {
    float x ;
    printf("entrer la note x\n");

```

```
scanf("%f", &x) ;
if (x < 0) printf(" erreur x < 0 ! \n");
else if (x < 10) printf(" non admis(e) \n") ;
else if (x < 12) printf(" passable \n");
else if (x < 14) printf(" assez bien \n") ;
else if (x < 16) printf(" bien \n") ;
else if (x < 18) printf(" tres bien \n") ;
else if (x <= 20) printf(" felicitations \n") ;
else printf(" erreur x > 20 ! \n");
return 0;
}
```

Correction de l'exercice [cos]

Le nombre d'itérations est connu, donc boucle "Pour".

```
#include <stdio.h>
#include <math.h>
// Affiche les sinus et les cosinus de 0 a 90 degres par pas de 10
int main () {
    int i ;
    float angle ;
    const float pi = 3.141592 ; // mot-cle const : pi n'est pas modifiable
    for(i=0;i<=90;i=i+10) {
        // attention : cos() et sin() prennent des double en argument,
        // et les valeurs doivent etre en radians
        angle = i * pi / 180 ; /* Regle de 3 */
        printf("cos(%f) = %f\n", angle,cos(angle));
        printf("sin(%f) = %f\n", angle,sin(angle));
    }
}
```

```
}  
return 0;  
}
```

Correction de l'exercice [Devinette]

L'algorithme utilisé pour ce problème est celui de la *recherche dichotomique* (l'intervalle de recherche est divisé par deux à chaque itération, et on garde la moitié qui contient la solution).

```
#include <stdio.h>  
  
// Devinette d'un nombre par dichotomie  
int main() {  
    int g, d, m ;  
    char r ;  
    printf(" Retenir un nombre entier entre 0 et 1000...\n") ;  
    printf(" ...puis appuyer sur une touche pour commencer.\n") ;  
    getchar(); // attente d'entree clavier  
    g = 0 ; d = 1000 ;  
    while ( g <= d ) {  
        m = ( g + d ) / 2 ;  
        printf(" Le nombre est-il egal a %d ? (o/n)\n", m);  
        scanf(" %c", &r) ;    /* NOTER l'espace avant %c */  
        if ( r == 'o' ) {  
            printf( " nombre trouve\n " );  
            return 1;  
        }  
        else {  
            printf("Le nombre est-il inferieur a %d ? (o/n)\n", m);  
            scanf(" %c", &r) ;    /* NOTER l'espace avant %c */
```

```
    }  
    if ( r == 'o' ) d = m - 1 ;  
    else g = m + 1 ;  
  }  
}
```

Remarque : une boucle du type “Répéter ... jusqu’à ce que” est aussi envisageable.