

## TD 3 - Environnement de programmation

### Techniques de Programmation

#### Exercice 1

Que fait le programme suivant ? Modifier les noms de variables et commentez le pour le rendre lisible.

```
#include<stdio.h>
int main() {
    int i,j,k=0,l,m,n;
    printf( "entrez les valeurs : \n" );
    scanf ( "%d%c",&i );
    scanf ( "%d%c",&j );
    if ( i>j ) { l=n=i;m=j; }
    else { l=n=j;m=i; }
    while ( l>=m ) { k++; l-=m; }
    printf ( " Resultat %d %d %d %d", i, j, k, l );
}
```

#### Exercice 2

Écrire un programme saisissant un tableau. La taille du tableau sera une constante nommée. Compiler ce programme de manière à obtenir le fichier après passage du préprocesseur. Visualiser le fichier obtenu. Compiler le programme obtenu et observer le fichier après compilation (transformation en assembleur). Assembler le fichier pour obtenir un binaire.

Nous allons maintenant introduire des options de précompilation dans le programme. Insérer une ligne de code vérifiant la bonne saisie de chaque élément du tableau. Cette trace pourra être supprimée ou affichée par changement d'une option de compilation. Utiliser pour cela les instructions de précompilation **#define**, **#ifdef** et **#ifndef**.

#### Exercice 3

On désire implémenter un système de navigation aérienne pour le contrôle aérien. Chaque avion est caractérisé par les informations suivantes : Code avion, altitude en pieds, cap en degré, profil (NIVEAU\_CONSTANT, EN\_MONTEE, EN\_DESCENTE) et inclinaison le cas échéant, autonomie, état (NOMINAL, ALERTE, DETRESSE).

Créer une ou plusieurs structures de données permettant de stocker informatiquement les informations caractéristiques d'un avion. Proposer un menu à un utilisateur permettant de créer un nouvel avion et de l'afficher.

ou

#### Exercice 4

Déclarer un entier **i** et un pointeur vers un entier **pi**. Initialiser l'entier à une valeur saisie au clavier. Faire pointer le pointeur sur **i**. Imprimer la valeur de **i** et la valeur de la variable pointée par **pi**.

Modifier la valeur de l'entier pointé par **pi** (en passant par le pointeur). Afficher la valeur de **i**.

## Exercice 5

Soient deux tableaux de flottants T1 et T2. Ils sont de taille 10. Écrire les instructions permettant de recopier dans T1, tous les éléments positifs de T2, en complétant éventuellement T1 par des zéros.

## Exercice 6

Copiez le tableau suivant et complétez-le pour chaque instruction du programme ci-dessous.

```
main()
{
    int A = 1;
    int B = 2;
    int C = 3;
    int *P1, *P2;
    P1=&A;
    P2=&C;
    *P1=(*P2)++;
    P1=P2;
    P2=&B;
    *P1--=*P2;
    ++*P2;
    *P1**=*P2;
    A=++*P2**P1;
    P1=&A;
    *P2=*P1/=*P2;
    return 0;
}
```

	A	B	C	P1	P2
Init.	1	2	3	/	/
P1=&A	1	2	3	&A	/
P2=&C					
*P1=(*P2)++					
P1 = P2					
P2=&B					
*P1--=*P2					
++*P2					
*P1**=*P2					
A=++*P2**P1					
P1=&A					
*P2=*P1/=*P2					

Implémenter le programme et tester la validité de vos résultats.

## Exercice 7

Reprendre le programme permettant de saisir un tableau d'entiers de taille fixe. Ajouter un menu proposant à l'utilisateur de :

1. saisir un nouveau tableau ;
2. afficher le tableau ;
3. afficher l'élément de rang *i* ;
4. rechercher un élément dans le tableau (la recherche se fera en utilisant un pointeur pour parcourir le tableau) ;
5. trier ce tableau par ordre croissant (le choix de l'algorithme de tri est libre - implémenter des fonctions de trace optionnelles lors du tri) ;