Université IBN KHALDOUN –TIARET- Faculté Des Mathématiques et de l'informatique
Département d'informatique, 2ière Année Licence : -2014/2015-

TD N°03 : Les pointeurs

Exercice 1 :

1. Quels sont les 2 principaux espaces de mémoire utilisées par un programme ?
2. Qu'est-ce qu'un pointeur ? Quels sont ces avantages ?
3. Quelle sont les quatre primitives de gestion du mémoire en langage C ?

Exercice 2:

1. Créer avec un pointeur de pointeur une matrice de k lignes et n colonnes à coefficients entier

Soit les deux structures de données suivantes :

```
struct Point {
char label ;
double x ;
double y ;
 };
```

```
typedef struct {
struct Point p1 ;
struct Point p2 ;
struct Point p3 ;
} Triangle ;
```

2. Quel se produit lorsque les commandes suivantes sont exécutées?

```
Triangle *tri = (Triangle *) malloc( sizeof(Triangle) );
Point * NewPoint = (struct Point *) malloc (sizeof (struct Point));
```

Exercice 3 :

Soit les programmes suivants :

(a) Quelle est la valeur de x :

```
int x=3;
int * px;
int y=-10;
int * py=&y;
px = &x;
*py = y + *px;
x = *py-x;
```

(b) Quelle est la valeur de i :

```
int i=0;
int *pi;
pi=&i;
*pi=*pi+1;
```

(c) Que afficher les trois instructions suivantes :

```
int tab[10]={5,8,4,3,9,6,5,4,3,8};
printf("%d \n",*tab);
printf("%d \n",tab[0]);
int *pt = tab;
pt++;
printf("%d \n",pt[0]);
```

Exercice 4 :

1. Quelle différence y a-t-il entre les deux programmes suivants ?

```
int main(){
  int i = 3, j = 4;
  int *p1, *p2;
  p1 = &i; p2 = &j;
  *p1 = *p2;

  return 0;
}
```

```
int main(){
  int i = 3, j = 4;
  int *p1, *p2;
  p1 = &i; p2 = &j;
  p1 = p2;

  return 0;
}
```

2. Etant données ces déclarations/initialisations, que valent les expressions dans le tableau ?

```
typedef struct toto{ int a; int b;} Toto;

int i1=1, i2=i1;

int *p1=&i1, *p2=&i2, *p3=&i2;
*p2=2;

Toto t1 = {1,2}, t2 = {1,2};

Toto *pt1=&t1, *pt2=&t2, *pt3=pt2;
```

(i1==i2)	
(p1==p2)	
(p2==p3)	
(*p2==*p3)	
(t1==t2)	
(t1.a==t2.a)	
(pt1==pt2)	
(pt2==pt3)	
(*pt1==*pt2)	

Exercice 5 :

Parmi les instructions qui suivent, lesquelles sont justes et lesquelles sont fausses

```
1. int *pa = 2;
2. *pa = 34;
3. int b = 4, *pb = &b;
4. *pb = 5;
5. int *pc;
6. printf("pc is equal to %d\n", pc);
7. printf("*pc is equal to %d\n", *pc);
8. pc = malloc(sizeof(int));
9. *pc = -2;
10. pa = pc;
11. free(pa);
12. pc = -4;
```

Exercice 6 :

Écrire un programme qui lit une chaîne de caractères CH de taille maximum 100 et détermine la longueur de la chaîne à l'aide d'un pointeur P.

Exercice 7 :

Ecrire un programme déterminant les nombres premiers qui figurent dans un tableau (statique). Pour ce faire, on conservera les nombres premiers dans un tableau (dynamique), au fur et à mesure de leur découverte