Université Mohammed V – Agdal
Faculté des Sciences – Rabat
Département d'Informatique

S. ZITI
2012/2013

Travaux dirigés 4
Programmation en Langage C
SMA3

Exercice 1 :

Les variables ***adr1*** et ***adr2*** sont des pointeurs pointant sur des réels. Le contenu de ***adr1*** vaut -45,78; le contenu de ***adr2*** vaut 678,89. Écrire un programme qui affiche les valeurs de ***adr1***, ***adr2*** et de leur contenu.

Exercice 2 :

1. Déclarer un texte comme un tableau statique initialisé par une chaîne de caractères constante, un tableau d'entiers statique pour compter les occurrences des lettres dont la taille est fixée par une constante et un pointeur pour parcourir le texte.
2. Initialiser le vecteur d'entiers avec un parcours par indice.
3. Compter les occurrences en utilisant la conversion entre le type char et le type int (la conversion d'un caractère donne son code dans le standard américain).
4. Afficher le résultat sur la sortie standard.

Exercice 3 :

Donner et expliquer le résultat de l'exécution du programme suivant :

```
#include <stdio.h>
#define taille_max 5
void parcours(int *tab)
{
t *q=tab;
  do
  {
    printf("%d:%d\n", q-tab, *q-*tab);
  }
  while (++q-tab < taille_max);
}
void bizarre(int **copie, int *source)
{
  *copie=source;
}
int main(void)
{
  int chose[taille_max] = {1,3,2,4,5}, *truc;
  printf("chose : \n");
  parcours(chose);
  bizarre(&truc, chose);
  printf("truc : \n");
  parcours(truc);
  return 0;
}
```

Exercice 4 :

1. Soit le code suivant

```
int a=3;
int b=10;
int c, *pa, *pb, *pc;
pa = &a;
*pa = *pa * 2;
pb = &b;
c = 3 * (*pb - *pa);
pc = pb;
pa = pb;
pb = pc;
```

À la fin de l'exécution, que valent *pa, *pb, a, b, et c ?

2. Les deux codes suivant sont-ils corrects, et si non pourquoi ?

```
{
int i, *p;
p = (int*) malloc(10 * sizeof(int));
for (i = 1; i <= 10; i++)
      p[i] = i * i;
}
```

```
{
int i, *p;
p = (int*) malloc(10 * sizeof(int));
for (i = 1; i <= 10; i++)
      p[i] = i * i;
}
```

Exercice 5 :

Écrire un programme qui lit deux tableaux A et B et leurs dimensions N et M au clavier et qui ajoute les éléments de B à la fin de A. Utiliser le formalisme pointeur à chaque fois que cela est possible.

Exercice 6 :

Écrire un programme qui lit un entier X et un tableau A de type int au clavier et élimine toutes les occurrences de X dans A en tassant les éléments restants. Le programme utilisera des pointeurs pour parcourir le tableau.

Exercice 7 :

Donner la trace du programme suivant :

```
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;
}
```