POINTEURS

Exercice 1

Les deux séries d'instructions suivantes sont-elles correctes? Pourquoi ?

```
int *ptr, nb=1;
*ptr=22;
ptr=&nb;
```

```
int n1,n2;
int *ptr=&n1;
ptr=10;
n2=ptr+2;
ptr=&n2;
```

Exercice 2

Un programme en C contient les instructions suivantes :

```
float a=0.001, b=0.003;
float c, *pa, *pb;

pa= &a;
*pa *= 2;
pb= &b;
c= 3*( *pb - *pa);
```

En supposant qu'un pointeur sur un nombre réel occupe 4 octets de mémoire que les nombres réels sont représentés selon la norme IEEE 754 et que l'adresse de la variable a est 1130 (en hexadécimal) :

- Quelle est la valeur assignée à &a ?
- Quelle est la valeur assignée à &c ?
- Quelle est la valeur assignée à &pb ?
- Quelle est la valeur assignée à pa ?
- Quelle est la valeur représentée par *pa ?
- Quelle est la valeur représentée par &*pa ?
- Quelle est la valeur assignée à pb ?
- Quelle est la valeur représentée par *pb ?
- Quelle est la valeur affectée à c ?

Exercice 3

Soit la partie de programme suivante :

```
int A = 1, B = 2, 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;
```

Copier le tableau suivant et le compléter pour chaque instruction du programme ci-dessus.

	A	B	C	P1	P2
p1=&A					
...					

Exercice 4

Soit les déclarations suivantes :

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};
int *P;
P = A;
```

Quelles valeurs ou adresses fournissent ces expressions ?

a) *P+2
b) *(P+2)
c) &P+1
d) &A[4]-3
e) A+3
f) &A[7]-P
g) P+(*P-10)
h) *(P+*(P+8)-A[7])

Exercice 5

Qu'affiche le programme suivant ? Pourquoi ?

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{   int *a, *s, i;
    s = a = (int *) malloc( 4 * sizeof(int));

    for (i=0; i<4; i++)
      *(a+i) = i * 10;

    printf("%d\n", *s++);
    printf("%d\n", (*s)++);
    printf("%d\n", *s);
    printf("%d\n", *++s);
    printf("%d\n", ++*s);
    free(a);
}
```

Exercice 6

Construire un programme C, permettant de gérer un tableau dynamique de nombres entiers. Considérer qu'au démarrage de l'application, le tableau dynamique ne contiendra aucune valeur et prévoir :

- le rajout d'un élément à la fin du tableau,
- le rajout d'un élément au début du tableau,
- l'insertion d'un élément en i-ème position,
- la suppression d'un élément à la fin du tableau,
- la suppression d'un élément au début du tableau,
- la suppression d'un élément de valeur donnée,
- l'affichage des valeurs du tableau.

Exercice 7

On dispose d'un tableau array contenant au plus 1000 mesures de températures exprimées en degré Celsius. On souhaite isoler dans des tableaux dynamiques :

- les mesures comprises entre -100 et -40 °C,
- les mesures comprises entre --40 et -20 °C,
- les mesures comprises entre -20 et -0 °C,
- les mesures comprises entre -0 et 10 °C,
- les mesures comprises entre 10 et 30 °C,
- les mesures comprises entre 30 et 100 °C,

Construire le programme C permettant :

- de saisir le tableau array,
- de ranger dans les 6 tableaux dynamiques les valeurs demandées
- d'afficher tous les résultats.

Exercice 8

On dispose de deux tableaux statiques contenant des caractères, délimités par le symbole '.' (le dernier caractère exploitable de ces tableaux est le caractère '.'). Construire un programme C permettant de fusionner le contenu de ces deux tableaux dans un troisième tableau, dynamique cette fois, de manière à ce que les caractères soient triés.

Bonus

Les tableaux associatifs comme ceux proposés en PHP n'existent pas en C. Les simuler (la clé pourra être un tableau de caractères, et la valeur un entier par exemple).