TD2 : Vie et mort des variables en mémoire (2/2)

Exercice 1 : Pointeurs et tableaux en C++, arithmétique des pointeurs

```
int monTab[] = {-25, -6, 8, 15, 38, 50, 72, 81, 98};

int * p = monTab;
```

Quelles valeurs ou adresses fournissent les expressions suivantes ?

a. *p+2

b. p+2

c. *(p+2)

d. &p

e. &monTab[4] - 3

f. monTab+3

g. &monTab[7] - p

h. *(p+8) - monTab[7]

Exercice 2 : Pointeurs et allocation dynamique de mémoire

Soit le programme C++ suivant :

```
#include <iostream>  /* entrées-sorties avec cin et cout */
using namespace std;

int main() {
  int e = 10;
  double r = 3.14;
  int * p1;
  double * p2;
  int i;

  p1 = new int [5];
  if (p1 == NULL) { cout << "Allocation ratee \n"; exit(1); }

  p2 = new double;
  if (p2 == NULL) { cout << "Allocation ratee \n"; exit(1); }

  for (i=0; i < 5; i++) { p1[i] = e-i; }
  *p2 = r;
  /* Faire la trace mémoire (1) */
  (*p1)*=5;
  *p2=p1[3]*2.0;
  /* Faire la trace mémoire (2) */
  e = 25;
  r = -8.3;
  /* Faire la trace mémoire (3) */
```

```
  delete [] p1;
  delete p2;
  /* Faire la trace mémoire (4) */
  return 0;
}
```

a. Expliquez ce que signifie l'instruction `p1 = new int [5];`

b. Proposez une autre façon d'écrire l'instruction `(*p1)*=5;`

c. Faites les quatre traces mémoire de ce programme, en supposant que la valeur de retour du main est stockée à l'adresse 3 566 758 966 et qu'il n'y a pas de problème d'allocation mémoire.

Exercice 3 : Appel à une fonction qui retourne un tableau

Dessinez l'état de la pile et du tas aux endroits indiqués en commentaires. Vous utiliserez le modèle théorique de pile vu en cours et au TD précédent. Vous supposerez que la valeur de retour du main est stockée à l'adresse 3 987 546 988 et qu'il n'y a pas de problème d'allocation dynamique de mémoire.

```
/* Précondition: tab1 et tab2 sont de même taille */
/* Postcondition: de la mémoire est allouée dans le tas, charge à l'utilisateur de la
                  libérer quand il n'en a plus besoin */
float * sommeTab(const float * tab1, const float * tab2, const int taille) {
  int i;
  float * resultat = new float [taille];
  for (i=0; i < taille; i++) resultat[i] = tab1[i] + tab2[i];
  return resultat;
  /* Faire la trace mémoire (1) */
}

int main() {
  float notes1[] = {8.5, 12.6, 14.5, 10.0, 9.1};
  float notes2[] = {12.2, 5.8, 17.3, 11.7, 7.6};
  float * somme = NULL;
  somme = sommeTab(notes1,notes2,5);
  /* Faire la trace mémoire (2) */
  delete [] somme;
  return 0;
}
```