TP 3 : Allocation dynamique de mémoire

Programmation en C (LC4)

Semaine du 12 février 2007

▶ Exercice 1

```
#include <stdio.h>

void affiche_vecteur(int *vecteur, int dimension) {
    int i;
    printf("( ");
    for (i = 0; i < dimension; i++)
        printf("%d ", vecteur[i]);
    printf(")\n");
}

void affiche_matrice(int **matrice, int lignes, int colonness)
{
    int i;
    for (i = 0; i < lignes; i++)
        affiche_vecteur(matrice[i], colonnes);
}
```

▶ Exercice 2

```
#include <stdlib.h>

int *alloue_vecteur(int dimension, int n) {
  int i;
  int *vecteur = malloc(dimension * sizeof(int));
  for (i = 0; i < dimension; i++)
    vecteur[i] = n;
  return vecteur;
}

void libere_vecteur(int *vecteur) {
  free(vecteur);
}
```

▶ Exercice 3

```
#include <stdlib.h>

int **alloue_matrice(int lignes, int colonnes, int val) {
    int i, j;
    int **matrice = malloc(lignes * sizeof(int *));
    for (i = 0; i < lignes; i++)
        matrice[i] = malloc(colonnes * sizeof(int));
    for (i = 0; i < lignes; i++)
```

```
        for (j = 0; j < colonnes; j++)
            matrice[i][j] = n;
    return matrice;
}

void libere_matrice(int **matrice, int lignes)
{
    int i;
    for (i = 0; i < lignes; i++)
        free(matrice[i]);
    free(matrice);
}
```

► **Exercice 4**

```
#include <stdlib.h>

int **alloue_matrice_zero(int lignes, int colonnes)
{
    int i;
    int **matrice = malloc(lignes * sizeof(int *));
    for (i = 0; i < lignes; i++)
        matrice[i] = calloc(colonnes, sizeof(int));
    return matrice;
}

int **genere_matrice_identite(int dimension)
{
    int i;
    int **identite = alloue_matrice_zero(dimension, dimension);
    for (i = 0; i < dimension; i++)
        identite[i][i] = 1;
    return identite;
}
```

► **Exercice 5**

```
int **alloue_matrice_pascal(int dimension) {
    int i, j;
    int **matrice = malloc(dimension * sizeof(int *));
    for (i = 0; i < dimension; i++)
        matrice[i] = malloc((i + 1) * sizeof(int));
    return matrice;
}

int **remplit_matrice_pascal(int dimension) {
    int i, j;
    int **matrice = alloue_matrice_pascal(dimension);
    for (i = 0; i < dimension; i++) {
        matrice[i][0] = 1;
        matrice[i][i] = 1;
    }
    for (i = 2; i < dimension; i++)
        for (j = 1; j < i; j++)
            matrice[i][j] = matrice[i - 1][j - 1] + matrice[i - 1][j];
    return matrice;
```

```
}

void affiche_matrice_pascal(int dimension) {
    int i;
    int **matrice = remplit_matrice_pascal(dimension);
    for (i = 0; i < dimension; i++) {
        affiche_vecteur(matrice[i], i + 1);
        free(matrice[i]);
    }
    free(matrice);
}
```

► **Exercice 6**

```
#include <stdio.h>

int *recupere_n_entiers(int n) {
    int i;
    int *tableau = alloue_vecteur(n, 0);
    for (i = 0; i < n; i++)
        scanf("%d", tableau + i);
    return tableau;
}

int *recupere_entiers(int n, int taille_max) {
    int i;
    int taille = n;
    int *tableau = alloue_vecteur(taille, 0);
    for (i = 0; i < taille_max; i++) {
        if (i >= taille) {
            taille += n;
            tableau = realloc(tableau, taille * sizeof(int));
        }
        scanf("%d", tableau + i);
    }
    return tableau;
}
```

► **Exercice 7**

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

int ***alloue_tableau_3D(int longueur, int largeur, int hauteur) {
    int i, j;
    int ***tableau = malloc(longueur * sizeof(int **));
    tableau[0] = malloc(longueur * largeur * sizeof(int *));
    tableau[0][0] = malloc(longueur * largeur * hauteur * sizeof (int));
    for (i = 0; i < longueur; i++) {
        tableau[i] = tableau[0] + i * largeur;
        for (j = 0; j < largeur; j++)
            tableau[i][j] = tableau[0][0] + (i * largeur + j) * hauteur;
    }
    return tableau;
}
```

```
void libere_tableau_3D(int ***tableau) {
    free(tableau[0][0]);
    free(tableau[0]);
    free(tableau);
}

void affiche_tableau_3D(int ***tableau, int longueur, int largeur, int hauteur) {
    int i, j, k;
    for (i = 0; i < longueur; i++) {
        printf("( ");
        for (j = 0 ; j < largeur; j++) {
            printf("( ");
            for (k = 0; k < hauteur; k++)
                printf("%d ", tableau[i][j][k]);
            printf(") ");
        }
        printf(")\n");
    }
}
```