TD de Système n° 1

Exercice 1 : Pointeurs

Compléter le tableau en indiquant les valeurs des différentes variables au terme de chaque instruction du programme suivant :

```
int main() {
  int a, b, c, *p1, *p2;
  a = 1, b = 2, c = 3;
  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;
}
```

a	b	c	p1	p2

Exercice 2 : pointeurs et arguments de fonctions

Écrire une fonction `swap` pour échanger les valeurs de deux variables de type `int`, et un `main` invoquant cette fonction.

Exercice 3 : manipulation de chaînes de caractères

Ecrire la fonction suivante :

```
int inverser_mots(char *texte)
```

Cette fonction suppose que `texte` est une chaîne de caractères formant un texte dont les mots sont séparés par des suites d'espaces arbitrairement grandes. Il peut y avoir ou non des espaces avant le premier mot, ainsi qu'après le dernier.

La fonction doit inverser l'ordre des lettres de chaque mot, mais sans changer l'ordre des mots :ainsi, `" cette phrase gz#! x "` sera transformé en `" ettec esarhp!#zg x "`. Le traitement devra se faire sur place, sans aucune déclaration de nouveau tableau.

Exercice 4 : allocation d'un tableau à deux dimensions

Proposer une représentation d'un tableau à deux dimensions qui stocke les entrées de façon contiguë en mémoire. Comment peut-on ensuite parcourir ce tableau sans utiliser la notation `[.]` ? Donner l'exemple du remplissage du tableau par des `0`, puis celui de l'affichage d'une ligne, et enfin de l'affichage d'une colonne.

Exercice 5 : manipulation de chaînes de caractères, arguments du main

Écrire un programme « `itineraire` » prenant en paramètre les références absolues de deux répertoires *rep1* et *rep2* d'une arborescence, et décrivant le plus court chemin entre eux. On ne se préoccupera pas de l'existence de *rep1* et *rep2*, et on supposera que les références données sont les plus simples (par exemple, pas de « `//` », de « `.` » ni de « `..` »).