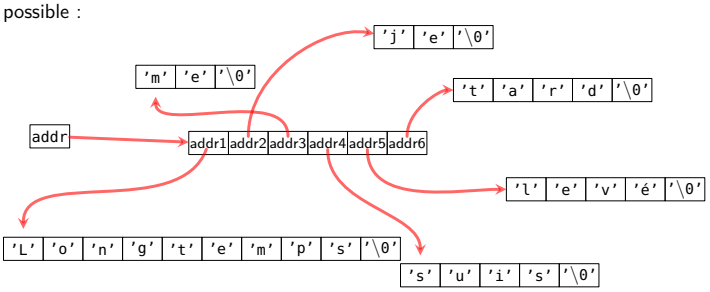
Notes

Algo/Prog – DUT 1

Notes

Tableau de tableaux ?

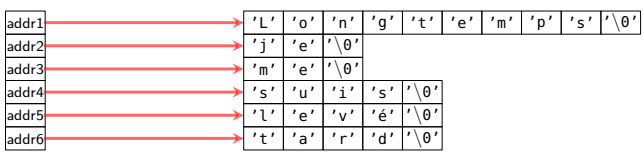
possible :



- voir TP 4 (`char ** t = lireFichierParMots("toto.txt",&x);`).

Notes

Illusion : bien « rangé »



- pratique d'y penser comme ça
- mais en fait pas de « localité » mémoire
- en C : tableau à double entrée $\neq$ tableau de tableaux
- on préférera le tableau de tableaux si rectangulaire

Notes

Tableau à double entrée I

```c
int t[4][3]; // exemple 1: tableau de 4 lignes chacune de longueur 3
int u = // exemple 2: tableaude 3 lignes, chacune de longueur 4.
  {
   {3, 5, 6  ,7},
   {7, 4, 3  ,2},
   {1, 9, 45 ,7}
  };
```

- toutes les lignes : même longueur
- longueur des lignes connue *statiquement* (à la compilation)
- en fait *un* tableau à une dimension

Notes

Tableau à double entrée II

```c
int t[4][3]; // exemple 1: tableau de 4 lignes chacune de longueur 3
t[1][2] = 2;
```

- `t[1][2] = 2;` équivalent à `t[ 1 * 3 + 2] = 2;`
- `t[i][j]` équivalent à `t[ i * largeur + j]`
- la largeur doit être connue à la compilation

Notes

Largeur connue à la compilation

- largeur déclarée dans les signatures de fonctions :

```c
void f(int t[][3]){
  ...
}
```

- Si plus de 2 dimensions : toutes les dimensions sauf la dernière

```c
void f(int t [][4][5]){
  ...
}
```

Notes

Morpion grill 3x3

```c
/* Affiche les neufs caractère comme une grille dans le terminal:
   3 lignes de 3 caractères */
void afficheGrille(char t[][3]){
  int i,j;
  for(i=0; i<3; i++){
    for(j=0; j<3; j++){
      printf("%c",t[i][j]);
    }
    printf("\n");
  }
}
```

boucle extérieure

boucle intérieure

Pattern typique :

- La boucle imbriquée (démo)

Notes

Démo boucles imbriquées

DÉMO

Notes

Notes

Notes

Notes