INF 111, L1, 2004/2005

Feuille 4

Septembre/Octobre 2004

Les tableaux, les procédures, les valeurs aléatoires

Laure Gonnord
http://www-verimag.imag.fr/~gonnord/

1 Petit cours

1.1 Tableaux

☺ Si on a besoin de 3 variables dans un programme, on en déclare 3, mais si on en veut 100 on fait comment ? La solution est un tableau de 100 cases.

- Pour déclarer le tableau, on fait `int tab[100];`
- Pour accéder à la $i$ième case, on fait : `tab[i]`, pour i entre `0` et `99` ($taille - 1$).

Attention, `i` et `tab[i]` ne désignent pas la même chose.

Exemple 1 *Déclaration/initialisation d'un tableau avec des cases toutes égales à 1*

```
#define N 10000

int main(void)
{
  int tab[N];
  for (i=0;i<N;i=i+1)
    {
      tab[i]=1;
    };
}
```

Exemple 2 *Impression du contenu du tableau tab, si il existe*

```
for (i=0;i<N;i=i+1)
    {
      printf("%d ",tab[i]);
    };
```

On peut fignoler en sautant à la ligne toutes les 10 cases, etc.

Remarque 1 Spécificité d'un tableau, quand il est en paramètre d'une fonction, on peut le modifier et les modifications sont enregistrées.

1.2 Valeurs aléatoires

☺ On peut obtenir des entiers tirés “au hasard" de la façon suivante :

- la fonction `int rand(void)` renvoie un entier tiré entre `0` et la constante `RAND_MAX`.
- la fonction `srand` permet de réinitialiser le générateur. On appelera cette fonction en faisant `srand(time(NULL));`.
- Il faut inclure la librairie `stdlib.h` (`#include<stdlib.h>`), ainsi que `time.h`.

EXEMPLE 3 *Génération aléatoire entre inf et sup :*

```
int hasard(int inf,int sup)
{
  int rando = rand();

  return (inf + (rando % (sup - inf +1)));
}
```

Explications :

- *On fait en sorte que le résultat de* `rand()` *(qui est situé entre* `0` *et une certaine constante* `RAND_MAX` *dont on se fiche de la valeur) soit ramené entre* 0 *et* $sup - inf$ *(compris), d'où le modulo.*
- *Ensuite, on rajoute* `inf` *pour avoir ce qu'on veut.*

1.3 Procédures

☺ Une procédure est une fonction qui ne retourne RIEN.
Une procédure est donc de la forme : `void toto(int a,int b)`, et ne comporte AUCUNE instruction return. Les procédures sont souvent utilisées pour les tableaux, car il est impossible de retourner un tableau à l'aide d'une fonction, on ne peut que le modifier.

2 Tableaux d'entiers

EXERCICE 1 *Pour chaque question, on s'attachera à bien faire la distinction entre procédure et fonction, et on fera attention aux bornes des tableaux.*

- *Déclarer une constante TAILLE, qui vaut 10000.*
- *Écrire une procédure* `void init_tab1(int tab[TAILLE])` *qui prend en argument le tableau* `tab` *(supposé connu) et qui le remplit de la valeur* 1.
- *Écrire une procédure* `void visu_tab(int tab[TAILLE])` *qui prend le tableau* `tab` *et l'affiche joliment.*
- *Écrire une procédure* `void init_tab2(int tab[TAILLE])` *qui prend en argument le tableau* `tab` *(supposé connu) et qui le remplit de valeurs aléatoires entre* 0 *et* $TAILLE - 1$.
- *Écrire une fonction* `int douze(int tab[TAILLE])` *qui RETOURNE* 1 *si le tableau comporte la valeur* 12, 0 *sinon. On utilisera une boucle for, et on fera attention à terminer dès que l'on trouve ladite valeur.*
- *Écrire une fonction* `int y_est_il(int el, int tab[TAILLE])` *qui RETOURNE* 1 *si le tableau comporte la valeur* `el`, 0 *sinon.*
- *Écrire une procédure* `void echange(int tab[TAILLE],int i,int j)` *qui échange les cases* i *et* j *du tableau* `tab`. *Faire attention à ne pas perdre de valeurs.*
- *Écrire une fonction* `int minitab(int tab[TAILLE)` *qui retourne le minimum du tableau* `tab` *(le minimum, pas l'indice du minimum). Faire la même chose avec le maximum.*

3 Tableaux 0-1

EXERCICE 2 *Faire le problème "énumération des parties d'un ensemble", annales d'examen, cahier (orange) de td, pages 49 à 53. En TP, on fera attention à tester toutes les fonctions une à une, en les appelant dans le programme principal.*