Licence 1ère année
Année universitaire 2006-2007

Le Langage C (INF123)

Série de TD n°5

Exercice 1

Considérons le programme suivant :

```
#include <stdio.h>
void fonction1(int) ;
int main()
{  int x=0, y=79 ;
   fonction1(x) ;
   if (y%x)
       printf("Ce ne sont pas des multiples\n") ;
   else
       printf("Ce sont des multiples\n") ;
   return 0 ;
}
void fonction1(int iArg)
{
   do
   { printf("Entrer un nombre : ") ;
     scanf("%d", &iArg) ;
   } while ( (iArg<=2) || (iArg>12) ) ;
}
```

1. En utilisant la représentation de la pile vue en cours, donner les états successifs de la pile lorsque les valeurs saisies au clavier sont 29 puis 4.
2. Expliquer pourquoi le programme ne fonctionne pas.
3. Corriger de deux manières différentes la fonction `fonction1` afin que le programme puisse fonctionner correctement.

Exercice 2

1. Donner l'affichage provoqué par le programme suivant :

```
#include <stdio.h>
void echange ( int, int) ;
int main()
{  int n=10, p=20, a=30 ;
   printf("avant appel   : %d %d\n", n, p) ;
   echange (n, p) ;
   printf("apres appel   : %d %d\n", n, p) ;
   return 0 ;
}
void echange ( int a, int b)
{  int n ;
   printf("debut echange : %d %d\n", a, b) ;
   n = a ;
   a = b ;
   b = n ;
   printf("fin echange   : %d %d\n", a, b) ;
}
```

2. Ecrire une fonction qui échange les valeurs entières pointées par deux pointeurs d'entier.
3. Ecrire une fonction qui échange les adresses contenues dans deux pointeurs d'entier.
4. Ecrire un programme C utilisant ces deux dernières fonctions.

Exercice 3

Un nombre premier est un entier strictement positif qui n'est divisible que par 1 et par lui-même.
Le crible d'Eratosthène est un procédé algorithmique pour établir la liste des nombres premiers inférieurs ou égaux à un certain entier donné n. On considère tout d'abord une liste L représentant les nombres entiers de 1 à n. On procède alors par étapes successives en "éliminant" de L tous les multiples de 2 (différents de 2), puis tous les multiples de 3 (différents de 3), de 5 (différents de 5), etc.
Le but de cet exercice est d'écrire un programme C permettant de mettre en oeuvre ce procédé. Pour cela, on considère un tableau de n entiers, initialement tous égaux à 1. (On remarquera que l'entier i correspond à l'élément d'indice i-1 dans le tableau.) Une étape d'élimination des multiples d'un entier k (voir l'exemple ci-dessous pour l'évolution des valeurs de k) consiste alors à affecter la valeur 0 aux éléments du tableau correspondant à des multiples de k (et qui sont différents de k). À la fin des étapes d'élimination, seuls les nombres premiers correspondront à des éléments de valeur 1 dans le tableau.
Exemple : déroulement du crible d'Eratosthène pour $n = 10$:

1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	1	0	1	0
1	1	1	0	1	0	1	0	0	0
1	1	1	0	1	0	1	0	0	0
1	1	1	0	1	0	1	0	0	0

Chaque ligne ci-dessus correspond à l'évolution du tableau au cours des étapes d'élimination. La première ligne correspond à l'initialisation, la deuxième à $k = 2$ et la troisième à $k = 3$. La quatrième ligne correspond à $k = 5$ qui est plus petit nombre premier strictement plus grand que 3. Le passage de $k = 3$ à $k = 5$ est effectué en remarquant que 5 est l'entier correspondant à la position du premier 1 parmi les éléments du tableau d'indice au moins 3. Enfin, la cinquième ligne correspond à $k = 7$. On s'arrête alors car tous les éléments d'indice au moins 7 sont nuls.

1. Ecrire une fonction qui retourne un entier strictement supérieur à 1 saisi au clavier.
2. Ecrire une fonction qui permet d'initialiser à 1 tous les éléments d'un tableau d'entiers.
3. Ecrire une fonction `MultipleDansTableau(int viTab[], int iN, int iK)` qui permet d'affecter la valeur 0 à tous les éléments du tableau `viTab` correspondant à des multiples de `iK` strictement plus grands que `iK` et inférieurs ou égaux à `iN`.
4. Ecrire une fonction `ProchainNombrePremier(int viTab[], int iN, int iP)` permettant de passer d'un nombre premier au nombre premier suivant. Cette fonction retournera le nombre premier (inférieur ou égal à `iN`) venant juste après le nombre premier `iP` dans le tableau `viTab[]`, ou 0 s'il n'y en a pas.
5. Ecrire une fonction `Eratosthene(int viTab[], int iN)` permettant de mettre en oeuvre le crible d'Eratosthène. Cette fonction modifiera les éléments du tableau `viTab` de manière à n'avoir à la fin de la fonction que les nombres premiers (entre 1 et `iN`) qui correspondent à des éléments de valeur 1 dans le tableau.
6. Ecrire une fonction `AfficheNombrePremier(int iN)` permettant d'afficher tous les nombres premiers compris entre 1 et `iN`. Cette fonction utilisera (au moins) les fonctions écrites aux questions 2 et 5.
7. Ecrire un programme C permettant de saisir au clavier un nombre entier strictement supérieur à 1 et d'afficher tous les nombres premiers compris entre 1 et ce nombre.