

Introduction au C TP1

1 Capacité des variables

1. Selon vous, que va-t-il se passer en exécutant ce code ?

```
int main(int argc, char argv[]) {
    short x = 1;
    while (x >= 1) {
        x = x + 1;
    }
    return 0;
}
```

2. Compilez et exécutez le code. Pour cela utilisez la commande :

gcc -Wall -ansi -o prog prog.c

gcc est le nom du compilateur que nous utilisons.

-Wall est une option de compilation qui vous permet d’avoir un grand nombre de diagnostics du compilateur ; il ne s’agira que de “Warnings”, mais lisez-les attentivement, ça peut vous permettre d’éviter bien des erreurs peu compréhensibles à l’exécution.

-ansi demande au compilateur de compiler en respectant le standard ansi.

-o permet de spécifier le nom de l’exécutable (par défaut il serait a.out).

Que se passe-t-il ? Vérifiez votre réponse auprès du professeur.

3. Pour comprendre ce qu’il se passe, affichez la valeur de x à chaque tour de boucle. Pour cela, utilisez la fonction printf. Pour savoir comment l’utiliser, vous pouvez lancer la commande *man 3 printf* dans un terminal.
4. La bibliothèque standard <limits.h> contient un certain nombre de constantes. Affichez les constantes SHRT_MAX et SHRT_MIN. À quoi correspondent ces deux constantes ?
5. Expliquez le comportement du programme. Vérifiez votre réponse auprès du professeur.

2 Pointeurs et tableaux

1. Exécutez le programme suivant.

```
#include <stdio.h>

void main(void) {
    int* p;
    printf("%d", *p);
}
```

Que se passe-t-il ? Que faudrait-il faire pour éviter ce problème ? Vérifiez votre réponse auprès du professeur.

2. Exécutez le programme suivant :

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

```
void f(){
    int nchiffre[20];
    for (i = 0; i < 25; ++i) {
        printf("nchiffre[%d]=%d\n", i, nchiffre[i]);
        nchiffre[i] = 0;
    }
}
```

```
void main(void) {
    f();
    printf("_fini\n");
}
```

Que se passe-t-il ?

3. Exécutez les programmes suivant :

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

```
void f(){
    int nchiffre[20];
    int i = 1010;
    printf("i=%d\n", nchiffre[-1]);
}
```

```
void main(void) {
    f();
    printf("_fini\n");
}
```

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

```
void f(){
    int i = 1010;
    int nchiffre[20];
    printf("i=%d\n", nchiffre[-1]);
}
```

```
void main(void) {
    f();
    printf("_fini\n");
}
```

Que se passe-t-il ?

4. Vérifiez vos réponses auprès du professeur.

3 Allocation de la mémoire et tableaux

Dans chacune des questions suivantes, vous devez tester le bon fonctionnement de votre fonction.

1. Écrivez une fonction qui multiplie par 2 la valeur d'un entier. La fonction ne doit rien renvoyer. Vérifiez votre réponse auprès du professeur.

2. Écrivez une fonction qui renvoie un pointeur vers un entier contenant la valeur 2022. Vérifiez votre réponse auprès du professeur.
3. Écrivez une fonction qui prend en entrée un nombre n et renvoie un tableau contenant les n premiers nombres de Fibonacci. Vérifiez votre réponse auprès du professeur.
4. Écrivez une fonction qui prend en entrée un tableau d'entier et sa taille et qui renvoie un tableau qui contient les valeurs du tableau d'origine suivi par les mêmes valeurs dans l'ordre inverse. Par exemple, si le tableau envoyé en entrée est $[1, 2, 3]$ la fonction renverra $[1, 2, 3, 3, 2, 1]$. Vérifiez votre réponse auprès du professeur.