

# CHAPITRE II

Q1: J'ai vu la conversion %i utilisée pour lire et écrire des entiers. Quelle est la différence entre %i et %d ?

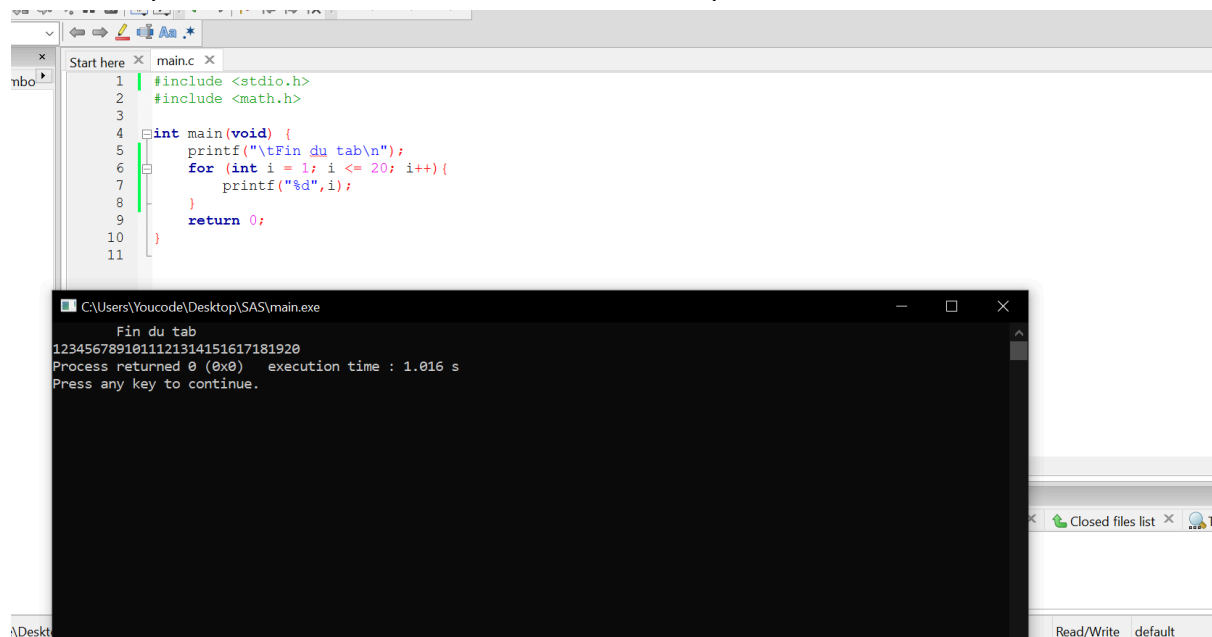
- + La différence entre %i et %d est que %i peut interpréter les entiers en formats décimal, octal et hexadécimal en fonction de l'entrée.

Q2: Si printf traite % comme le début d'une spécification de conversion, comment puis-je imprimer le caractère % ?

- + On peut imprimer "%" lui-même par cette expression : "%%".

Q3: La séquence d'échappement \t est censée faire avancer printf jusqu'au prochain taquet de tabulation. Comment puis-je savoir à quelle distance se trouvent les taquets de tabulation ?

- + On peut utiliser ce code là, Donc \t saute 10 espaces.



The image shows a screenshot of a C program being compiled and executed. The code is in a file named `main.c` and is as follows:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void) {
5     printf("\tFin du tab\n");
6     for (int i = 1; i <= 20; i++){
7         printf("%d", i);
8     }
9     return 0;
10 }
11
```

The execution output in the terminal window is:

```
C:\Users\Youcode\Desktop\SAS\main.exe
    Fin du tab
1234567891011121314151617181920
Process returned 0 (0x0)   execution time : 1.016 s
Press any key to continue.
```

Q4: Que fait scanf s'il lui est demandé de lire un nombre mais que l'utilisateur entre une entrée non numérique ?

- + Dans ce cas, "scanf" donne 0 comme une valeur de ce input.

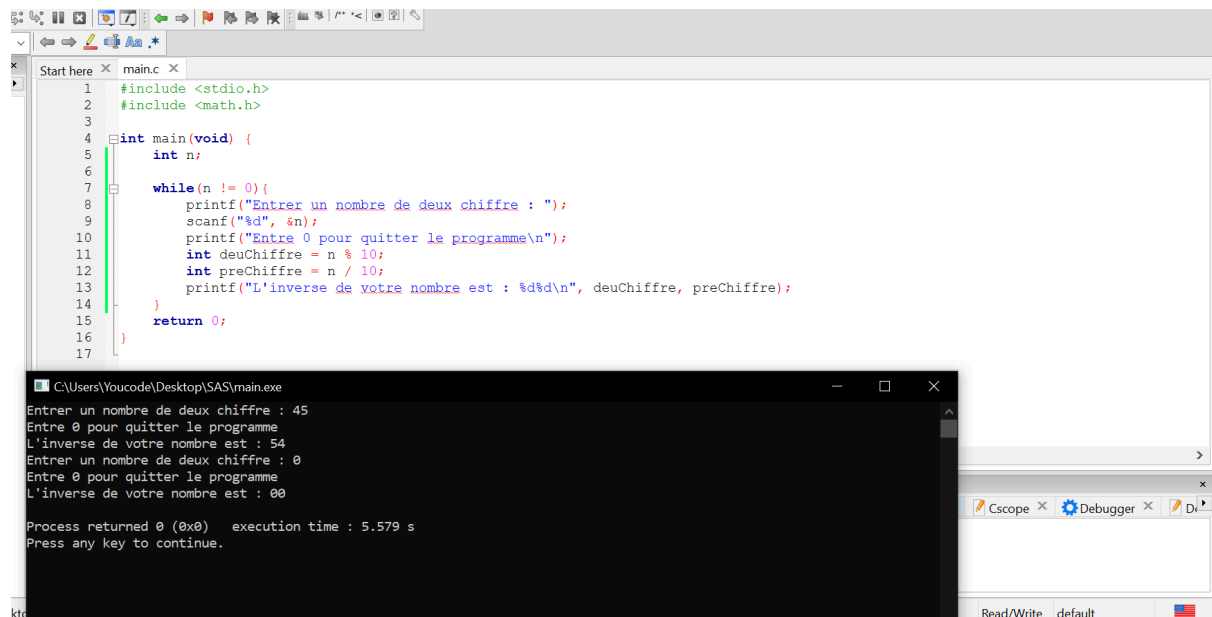
Q5: Je ne comprends pas comment scanf peut "remettre" des caractères et les relire plus tard.

Q6: Que fait scanf si l'utilisateur met des signes de ponctuation (des virgules, par exemple) entre les nombres ?

- + Elle prend le nombre qu'est après le premier signe de ponctuation.

## CHAPITRE III

2)



The image shows a C program in Visual Studio Code and its execution output in a terminal window. The program is named `main.c` and is located at `C:\Users\Youcode\Desktop\SAS\main.c`. The code is as follows:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void) {
5     int n;
6
7     while(n != 0){
8         printf("Entrer un nombre de deux chiffre : ");
9         scanf("%d", &n);
10        printf("Entre 0 pour quitter le programme\n");
11        int deuChiffre = n % 10;
12        int preChiffre = n / 10;
13        printf("L'inverse de votre nombre est : %d%d\n", deuChiffre, preChiffre);
14    }
15    return 0;
16 }
17
```

The terminal window shows the execution of the program. It prompts the user to enter a two-digit number. The user enters 45, and the program outputs the inverse number, 54. The user then enters 0, and the program outputs the inverse number, 00. The program then returns 0 (0x0) and the execution time is 5.579 s. The terminal window also shows the prompt "Press any key to continue."