PW sheet No. 2

[Algorithmics and data structures 1]

Part 1: Input statement

Input a number on the keyboard

The following program allows you to read and store a number in a variable ***x***. This number is a real number (***float*** data type). The variable ***x*** is then re-displayed using the ***printf*** instruction.

```
#include <stdio.h>                        /* to be able to read and write */
int main()                                                /* Main Program */
{
float x;                        /* Declaring a variable x (real number) */
printf("Please enter a real number on the keyboard\n");
scanf("%f", &x);                /* read the value of x from the keyboard */
printf("You typed %f, Goodbye", x);                      /* displaying x */
return 0;
}
```

a. Create a new project.

b. Type this code, then compile and run.

c. Replace *float* with ***int***, then compile and run. What do you conclude? Fix it so that it can enter and display an integer.

Concluding remarks

1. Reading in the C programming language is done using the ***scanf statement***, which is written as follows

   scanf("format d'affichage", &variable1);

2. The principle is similar to ***printf()***: you use a formatting string and a sequence of arguments. The difference is that here, the arguments are addresses. To read an integer, you will do the following:

   ```
   int i;
   scanf("%d", &i);
   ```

   The & operator tells the function that we are passing the address of the variable ***i***, not the variable itself.

3. Multiple values can be entered at once. The user must separate them with a carriage return, a space, or a tab:

   ```
   int i,j,k;
   scanf("%d%d%d",&i,&j,&k);
   ```

4. scanf uses the same formatting codes *(d, u, o, x, X, c, f, e, E)* and modifiers *(h, l, L)* as printf.

Part 2: The assignment statement

Swapping two variables

We asked a first-year computer science student to write a program to swap the values of two variables ***x*** and ***y***. He proposed the following program:

```
int main(void)
{
float x, y ;
printf("Enter x : ");
scanf("%f", &x);
printf("Enter y : ");
scanf("%f", &y);
x=y;
y=x ;
printf("The new value of x is  %f.\n", x);
printf("The new value of y is  %f.\n", y);
return 0;
}
```

a. Create a new project.
b. Type this code, then compile and run.
c. Correct this program so that it actually performs a permutation between two real numbers.

Concluding remarks

1. Assignment is the operation of assigning a value to a variable. In C, it is written as follows:

Variable_name = expression ;

The "=" symbol in the assignment has a completely different meaning to mathematical equality.

2. ***expression*** can be a constant or a variable or an evaluable expression.
3. The value of the expression must be of the same type as the variable

Part 3: Important notes

Note 1: Identifiers

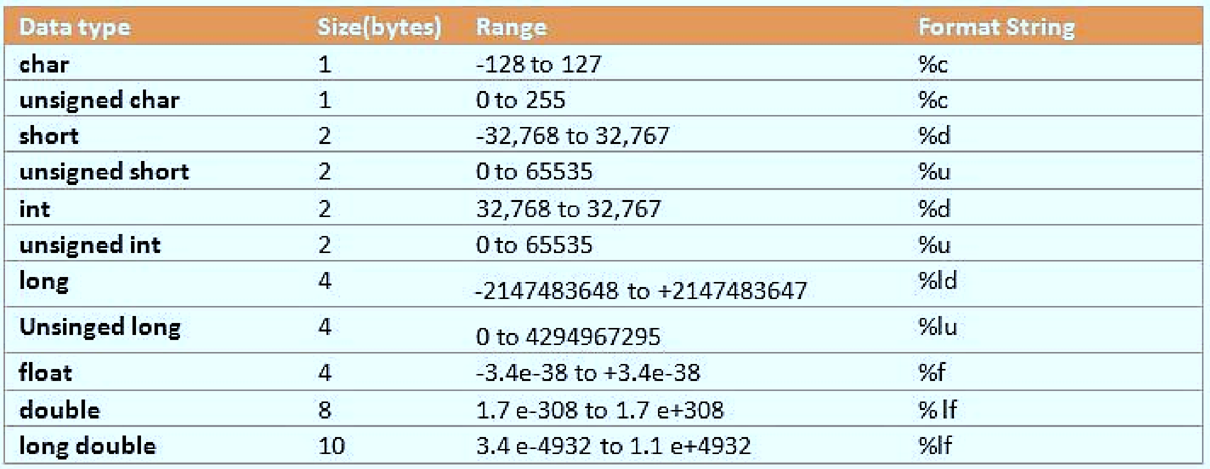
The use of identifiers must meet a number of requirements:

- Identifiers can be composed of letters, numbers, and the underscore character (_).
- Identifiers must start with a letter.
- The first 32 characters (or 8 characters, in some cases) are significant.
- Identifiers are case-sensitive.
- Identifiers cannot be reserved words.

Examples :

solution1 is a valid identifier (consisting of letters and 1)
1solution is not a valid identifier.
unit price is not a valid identifier (it contains a space).
unit_price is a valid identifier.
day, Day and DAY are 3 different identifiers.
int is not a valid identifier. This is a keyword used in C.

Note 2: Simple Predefined Types in C



Data type	Size(bytes)	Range	Format String
char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
short	2	-32,768 to 32,767	%d
unsigned short	2	0 to 65535	%u
int	2	32,768 to 32,767	%d
unsigned int	2	0 to 65535	%u
long	4	-2147483648 to +2147483647	%ld
Unsinged long	4	0 to 4294967295	%lu
float	4	-3.4e-38 to +3.4e-38	%f
double	8	1.7 e-308 to 1.7 e+308	% lf
long double	10	3.4 e-4932 to 1.1 e+4932	%lf

Part 4: Application Exercises

1. Write a program that reads an integer from the user and displays its double and triple.
2. Write a program called Rectangle that allows the user to enter the length and width of a rectangle, and calculates and displays the perimeter and area of the rectangle.
3. Write a program that prompts the user to enter a distance in miles and converts it to feet. Knowing that 1 mile = 1.609 kilometres and 1 metre = 3.2809 feet.
4. Write a program called ***Average*** that prompts the user to enter three grades, ***grade1***, ***grade2***, and ***grade3***, and then calculates and displays their average.
5. Write a program called ***Operations*** that prompts the user to enter two integers and then prints their sum, difference, product, and quotient.
6. Develop a program called ***Conversion*** that converts a time entered from the keyboard into a duration in minutes. The program should prompt the user to enter the hours and minutes, then calculate and print the total number of minutes.
7. Write a program that calculates the equivalent resistance Req of three resistances R1, R2 and R3 connected in parallel. Knowing that:

$$\frac{1}{\mathrm{Re}\,q} = \frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3}$$

8. Implement a program called ***DistanceBetweenPoints*** to calculate the distance between two points whose coordinates are provided by the user.
9. During a promotion, a store offers a 10% discount on all products. Write a program called "***Reduction***" that allows the user to enter the price of a product and displays the amount of the discount and the new price calculated after the discount..
10. Write an algorithm that performs a circular permutation of the integer values of three variables x, y, z (i.e. the value of y in x, the value of z in y and the value of x in z).