



TP1 : Variables and Basic Instructions

Ex 1.1 : Some display formats

1. Create a new project.
2. Copy this program into the main.c

```
#include <stdio.h>
#include <stdlib.h>
int main( )
{ int a=2 ,b=-5, c=1234 ,d=-5678;
  double x=1.2345, y=123.45 ,z=5.6e4;
  printf("a=%3d \n" , a);
  printf("b=%5d \n" , b);
  printf("c=%5d \n" , c);
  printf("d=%5d \n" , d);
  printf("x=%10lf \n" , x);
  printf("x=%10.3lf \n" ,x);
  printf("y=%10lf \n" , y);
  printf("y=%10.2e \n" , y);
  printf("z=%8.2lf \n" , z);
  printf("z=%e \n" , z);
  return 0; }
```

3. Compile and run your project.
4. Analyse the results obtained.

Ex 1.2 : Identifiers

1. Which of the following identifiers are accepted by the C language for calling variables?
(Explain why not)

- | | | | | |
|---------------|-------------|-----------------|--------------|-----------|
| 1) age | 2) Var1 | 3) _BAC_AVERAGE | 4) N°tel | 5) double |
| 6) FinalScore | 7) 3numbers | 8) Square-root | 9) Algo Note | 10) café |

2. Declare integer variables using the identifiers above (e.g: `int age ;`), then compile the program. Compare your answer to the Code::Blocks compiler's and correct any errors.

3. Initialize each of these variables (e.g: `int age=19 ;`).
4. Add the display instruction "`printf`" (e.g: `printf ("% d", age) ;`)
5. Compile and run the program.
6. Modify the values of these variables through assignments in the program (e.g: `age = 21 ;`)
7. Then display the values of these variables with "`printf`"

Ex 1.3 : Take a guess !

Consider the following program:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main()
4 { int x, y; // tow integer type variables.
5 double z; // variable of real type.
6 printf ( "Give the value of x: \n" ); // .....
7 scanf ( "%d" , &x ); // .....
8 printf ( "Give the value of y: \n" ); // .....
9 scanf ( "%d" , &y ); // .....
10 z = x/y; // .....
11 printf ( "La valeur de z est : %f \n" , z ); // .....
12 return 0; }
```

1. Complete the various comments before each instruction.
2. Guess what the above program does.
3. Run this program with the following values: $x=15, y=2$.
4. What do you notice?
5. Replace line10 with the following assignment : $z=(double)x/y$;
6. Compile and run.
7. Conclude.

Ex 1.4 : Assignments

Consider the following program:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main()
4 { int x, y, z ;
5 x=5;
6 y=3;
7 z=x+y;
8 x=2 ;
9 y=y+z ;
11 z=y-x ;
10 printf ( " x=%d \t y=%d \t z=%d \n " , x, y, z );
12 return 0;
13 }
```

1. Show execution history of this program.
2. Create a new project and copy this program into the main.c
3. Compile and run your project.
4. Compare your execution history with the result obtained.

Ex 1.5: Length

Ask the user to enter the length and width of a rectangle in meters. Then calculate the length of the diagonal and convert it into centimeters and millimeters. Display the three values with their appropriate units.

Ex 1.6: Ideal weight

To calculate my ideal weight, I chose the formula of *Monnerot-Dumaine*:

$$\mathbf{Ideal_Weight = (Height - 100 + (4 * Circumference_wrist))/2}$$

which takes into account bone and muscle mass by introducing the circumference of the wrist. Write a program that calculates the ideal weight, noting that the height and circumference of the wrist are expressed in *cm* and the weight is calculated in *kg*.

Ex 1.7: Percentage

Write a program that:

1. Asks the user to enter the net price of an item and a percentage of VAT (type: int) to be added.
2. Calculates the value corresponding to this percentage of the net price **VAT**.
3. Calculates and displays the total price using the following formula: (Beware of automatic type conversions.)

$$\mathbf{TotalPrice = NetPrice + VAT}$$

Notes: provide clear messages for user input, and display results with appropriate units.

Ex 1.8: Sum

Write a program that calculates the sum of four integer numbers entered by the user.

1. Using 5 variables.
2. Using only 2 variables.