Lesson 62(Find a number)

If we have an array that contains values, and we want to search for a certain number.

```
#include <stdio.h>
int main() {
int i, y, x[] = { 8,5,6,90,3,1 };
printf("Enter a number: ");
scanf("%d", &y);
for (i = 0; i < 6; i++)
if (x[i] == y)
printf("Found\n");
else
printf("Not Found\n");

}
```

Try the code yourself : Click Here!

If the user enters a number, he will pass on each number and execute what is in the **if** command. If in this roll the condition is **false** and the number does not exist, it will print Not found in and this means that if it does not find the number in 5

cycles, it will print **not found**, and if it finally finds it at the end of the Array, it will print **Found**.

When we find a print command that has printed a large number, and we need one print command, it is an order to print inside **for** command.

input:

Enter a number: 1

output:

Not Found

Not Found

Not Found

Not Found

Not Found

Found

There are some people who use **return 0** or **break**:

```
#include <stdio.h>
int main() {
int i, y, x[] = { 8,5,6,90,3,1 };
printf("Enter a number: ");
scanf("%d", &y);
for (i = 0; i < 6; i++)
if (x[i] == y){
```

```
printf("Found\n");
return 0;
}
else
printf("Not Found\n");
}
```

input:

8

output:

found

Try the code yourself : Click Here!

There is **for** command, and for will spin on each element. If you don't find it, it will print not found until you find it

```
#include <stdio.h>
int main() {
int i, y, x[] = { 8,5,6,90,3,1 };
printf("Enter a number: ");
scanf("%d", &y);
for (i = 0; i < 6; i++)
if (x[i] == y) {
printf("Found\n");
return 0;
}
```

Try the code yourself : Click Here!

This program works in one case. If the number you want to search for is not here, it is considered that we have canceled the **for** command because it will rotate only once if it finds it or does not find it. Therefore, the correct solution for this program is that we output the print command outside f**or** and make a variable that counts the number If he finds it he adds one

And it will be like this:

```
#include <stdio.h>

int main() {

int i, y, x[] = { 8,5,6,90,3,1 },found=0;

printf("Enter a number: ");

scanf("%d", &y);

for (i = 0; i < 6; i++)

found = x[i] == y;

//if the number exists then true

//if found equals 1 -> the number exists

/*

if (found > 0)

printf("found");

else

printf("Not found");

*/
```

```
printf(found ? "Found" : "Not found");
/*
if (found)
printf("Found\n");
else
printf("Not Found\n");
*/
}
```

input:

```
1
```

output:

```
found
```

Try the code yourself : [Click Here!]

Or :

```
#include <stdio.h>
int main() {
int i, y, x[] = { 8,5,6,90,3,1 };
printf("Enter a number: ");
scanf("%d", &y);
for (i = 0; i < 6; i++)
```

```
if (x[i] == y){
printf("Found\n");
return 0;
}
printf("Not Found\n");
//If the condition is not true, it prints not found
}
```

The command return 0 ends the program, so any code after the return won't work.

Try the code yourself : Click Here!