Lesson 78 sprintf

The most important thing in programming is how you deal with orders, not how much you memorize orders.

This command is used if we have defined characters of an array, but we did not put a name in it and we want after a certain number of codes to add words to record in this Array.

Example:

```
#include <stdio.h>

int main() {

char x[100] = "Programming is fun";

printf("%s\n", x);

}
```

output:

Programming is fun

Try the code : Click Here!

Here it is printed normally because we have recorded the speech in the characters of the array, but what if we don't?

```
#include <stdio.h>

int main() {

char x[100];

x = "Programming is fun";

//error shows up here
```

```
printf("%s\n", x);
}
```

Try the code : Click Here!

How to correct this, we can use **strcat** :

```
#include <stdio.h>
#include <string.h>
int main() {
char x[100];
x [0]= '\0';
//x is empty now
strcat(x, "Programming is fun");
//now we move the words into x
printf("%s\n", x);
}
```

output:

Programming is fun

Try the code : Click Here!

We can shorten the lines and use the sprintf command

As follows :

We write the **sprintf** command, then ( ), then the **name of the string** in which we will transfer the speech, then a comma,

Then "" and within them the words to be written:

```
#include <stdio.h>
int main() {
char x[100];
sprintf(x, "Programming is fun");
//add the text to x
printf("%s\n", x);
}
```

output:

Programming is fun

Try the code : Click Here!

printf : prints on the **screen**

fprintf : prints in a **file**

sprintf: prints in a **String**

But there is another difference, which is that in the **sprintf** command, if we use it for the same string more than once, it **deletes** the recorded text and adds the new one to be added.

Example :

```
#include <stdio.h>
int main() {
char x[100];
sprintf(x, "Programming is fun");
```

```
sprintf(x, "Hello Gammal Tech");
//add the text to x
printf("%s\n", x);
}
```

output:

Hello Gammal Tech

Try the code : Click Here!