Unconditional branching

Unconditional branching is when the programmer forces the execution of a program to jump to another part of the program. Theoretically, this can be done using a good combination of loops and if statements. In fact, as a programmer, you should always try to avoid such unconditional branching and use this technique only when it is very difficult to use a loop. Consider the following example:

```c
#include <stdio.h>
#define ARRAY_SIZE 10

int main(void)
{
    puts("Hello 1");
    goto LAB1;
    puts("Hello 2");
    LAB1:
    puts("Hello 3");
    return 0;
}
```

The above example illustrates a label `LAB1` and the keyword `goto` that is used to change the direction to of the program flow. The following example illustrates a different way of looping 10 times using a `goto` statement instead of a for loop:

```c
#include <stdio.h>
#define ARRAY_SIZE 10

int main(int, char **)
{
    int i = 0;
    AGAIN:
    puts("Hello");
    i++;
    if(i < 10)
    {
        goto AGAIN;
    }
    return 0;
}
```
Although the above program will do exactly the same thing as a for loop does, it is highly recommended that you DO NOT USE this technique. A lot of goto statements in a C program will make the program unstructured and difficult to trace. Sometimes this is called spaghetti programming.