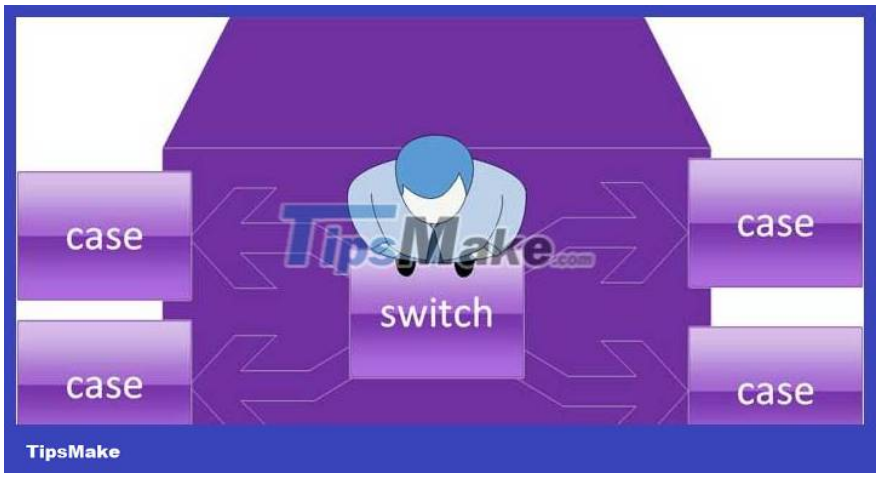
# Switch in C

Instead of writing multiple if..else statements, you can use the switch statement in C. The Switch statement in C selects one of many blocks of code to be executed.

Instead of writing many **if.else statements** , you can use **switch statements in** C. The Switch statement in C selects one of many blocks of code to be executed.




## Basic syntax of Switch in C

```
switch(expression) { case x: // code block break; case y: // code block break; d
```

Here's how it works:

1. The expression `switch`is evaluated once.
2. The value of the expression is compared with the value of each case.
3. If there is a match, the relevant block of code will be executed.
4. The command `break`exits the switch block and stops execution.
5. The command `default`is optional, specifying some code to run if there is no matching case.

The example below uses the number of days of the week to calculate the day of the week:

For example:

```
#include int main() { int day = 4; switch (day) { case 1: printf("Monday"); breal
```

## Break keyword

When C reaches the keyword `break`, it will exit the switch block. This action will stop further code deployment and test cases inside the block. When a suitable case is found, it means the job is complete. Now is the time to rest, no need for further testing.

A break can save a lot of implementation time because it skips execution of all the remaining code in the switch block.

## Default keyword

Keywords `default`specify some code to run if there is no matching case. For example:

```
#include int main() { int day = 4; switch (day) { case 6: printf("Today is Satur
```

Note: The keyword `default`must be used as the last command in `switch`and it does not require a break.

Above are **the things you need to know about Switch in** C. Hope the article is useful to you.

You finished reading the article "**Switch in C**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.