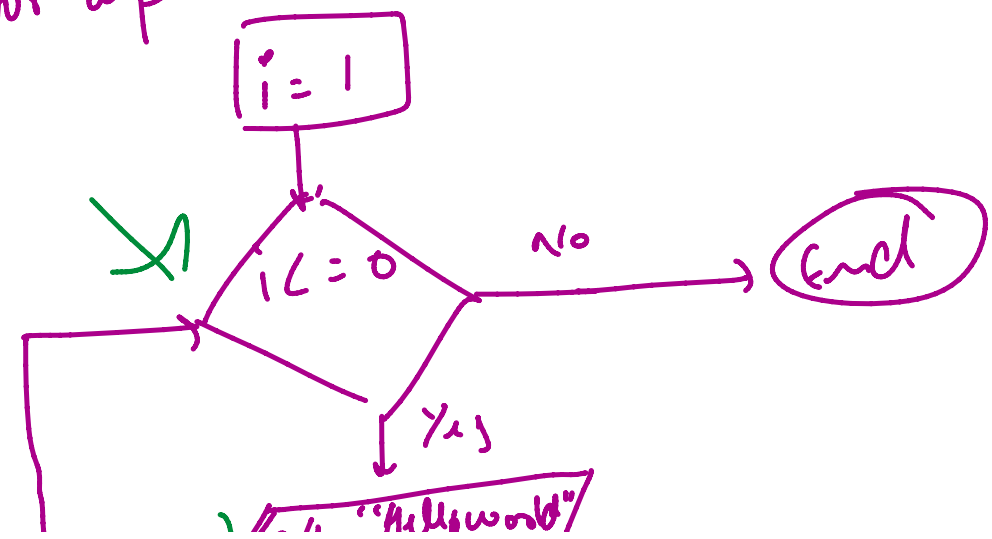
while loop, for loop:

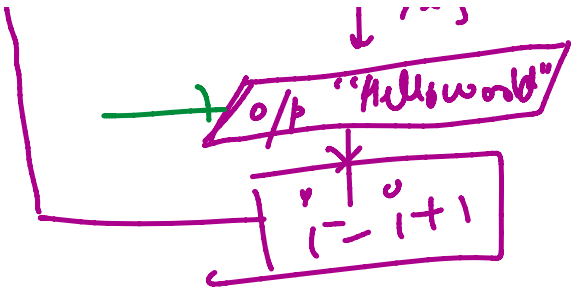
do while loop: syntax ↓

```
do {
    // tasks
    // updation
} while (condtn) ;
```

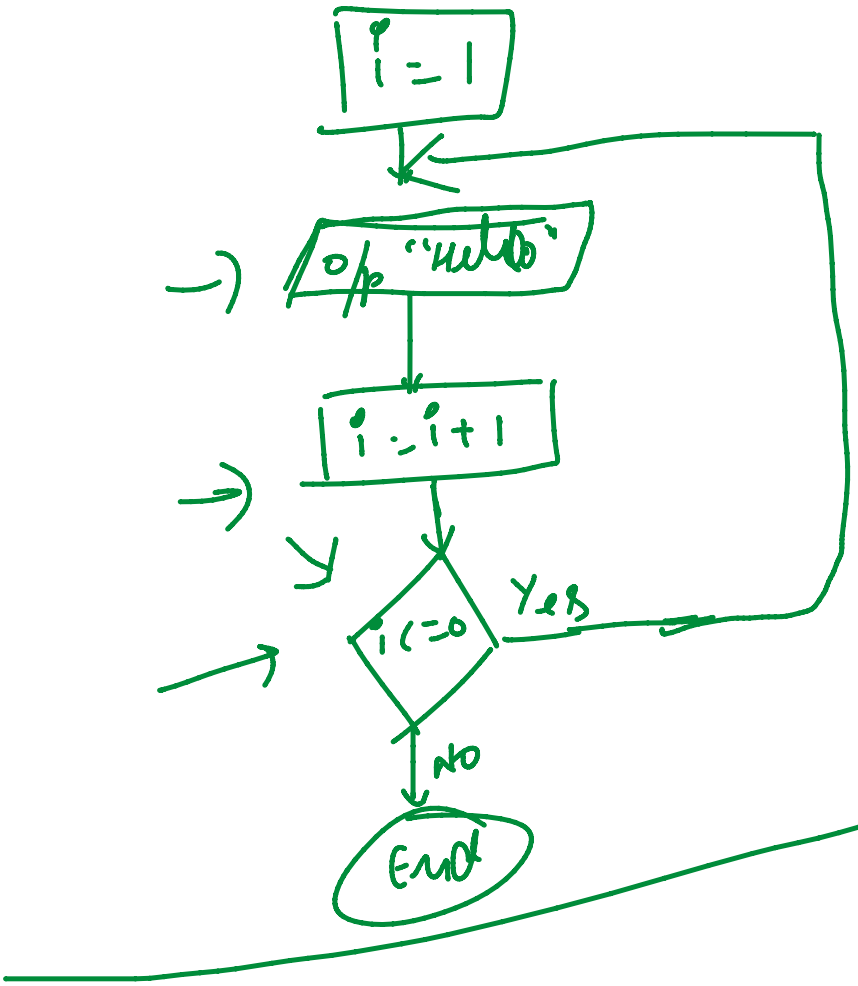
make a flow chart of [while/ for] loop and do-while loop.

while/ for loop.





do - while loop.



switch-case :- Syntax.

switch ( ) {

variable that is switching values (int, char)

case :

value it is taking

c :

is true)

case :

default :

}

It can make our code more structured & readable & a little faster than if-else ladder.
And also it can reduce lines of code in our program.

whatever you can do with switch-case you can do with if-else.
But it is not necessary that you can do everything with switch case that you can do with if-else..

Ternary operators. (? :)

It is a substitute for if-else

Syntax

(expr/condition) ? (statement 1 / task 1) : (task 2 / st. 2)

if ( ) ? (if wala task) : (else wala task)

task)

Adv. of Tertiary operators:-

→ Easy to write, small code / syntax.

→ You can nest it.

Disadv.

→ complex code. (the more nested it becomes the more harder it is for compiler to convert into binary)

→ Is slow compared to if else (if nested 2/3 times that it may be faster or equal to if-else but after that it is slower than if else)

→ Hard to debug.