



Conditional Statements

- The if statement
- The switch statement



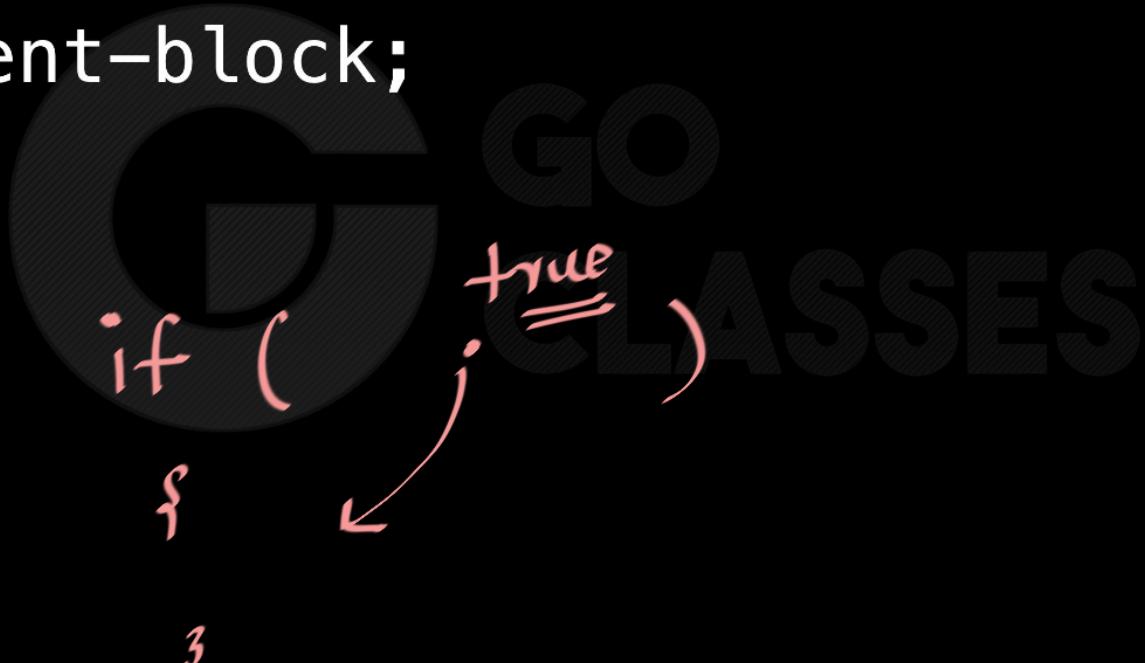
GO If – Else Statements



C Programming

```
if (test expression)
{
    statement-block;
}
```

A pink curved arrow points from the word "test" in the first line of the code to the handwritten underlined word "condition" to its right.





C Programming

```
int target = 10; target is true
if (target == 10) {
    printf("Target is equal to 10");
}
```

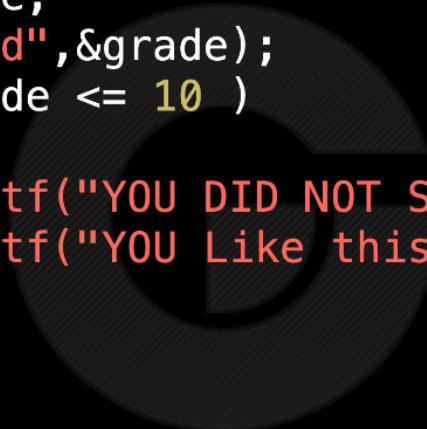
10

target



C Programming

```
#include<stdio.h>
int main()
{
    int grade;
    scanf("%d",&grade);
    if ( grade <= 10 )
    {
        printf("YOU DID NOT STUDY.\n");
        printf("YOU Like this course ! \n");
    }
}
```



GO
CLASSES





C Programming

```
#include<stdio.h>
int main()
{
    int grade;
    scanf("%d",&grade);
    if ( grade <= 10 )
        printf("YOU DID NOT STUDY.\n");
    printf("YOU Like this course ! \n");
}
```



```
#include<stdio.h>
int main()
{
    int grade;
    scanf("%d",&grade);
    if ( grade <= 10 )

        printf("YOU DID NOT STUDY.\n");
        printf("YOU Like this course ! \n");
}
```

```
#include<stdio.h>
int main()
{
    int grade;
    scanf("%d",&grade);
    if ( grade <= 10 )
    {
        printf("YOU DID NOT STUDY.\n");
        printf("YOU Like this course ! \n");
    }
}
```



C Programming

```
#include<stdio.h>
int main()
{
    int grade;
    scanf("%d",&grade);
    if ( grade <= 10 )
        printf("YOU DID NOT STUDY.\n");
        printf("YOU Like this course ! \n");
}
```

If there are no braces ({}) following the if statement then only the next immediate statement belongs to the if statement.



C Programming

```
#include<stdio.h>
int main()
{
    if(0)
        printf("statement 1\n");
    printf("statement 2\n");
    printf("statement 3\n");
    return 0;
}
```

false

0 ⇔ false

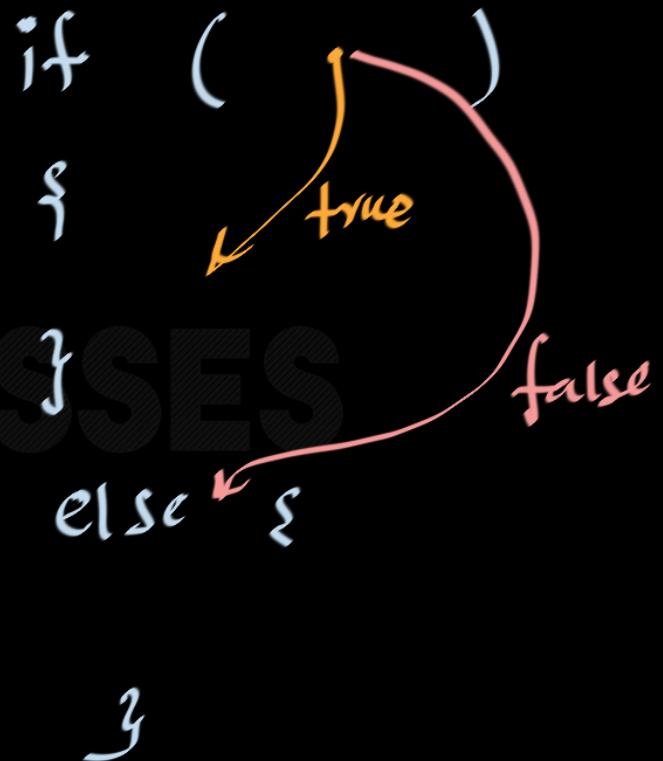
anything other
than 0

⇒ true

The **else** Clause

```
#include <stdio.h>

int main() {
    int i = 20;
    if (i < 15){
        printf("i is smaller than 15");
    }
    else{
        printf("i is greater than 15");
    }
    return 0;
}
```





Nesting if.. else

```
if ( num > 0 )      // Outer if
    if ( num < 10 )  // Inner if
        printf("num is between 0 and 10");
```

```
if ( num > 0 )      // Outer if
{
    if ( num < 10 )  // Inner if
        printf("num is between 0 and 10");
```

{





Nesting if.. else (Continue..)

```
if ( num > 90 )
{
    printf( "You earned an A" ) ;
}
else
    if ( num > 80 )
    {
        printf( "You earned a B" ) ;
    }
```





Matching if... else parts

```
if(a<10)
```

```
if (a % 2 ==0)
printf("a is even and less than 10\n");
```

```
else
```

```
printf("mystery");
```





C Programming

```
int i = 1, j = 5, k = 4, number;
```

```
if(i>j)
    if(i>k)
        number = i;
else
    number = k;
else
    if(j>k)
        number = j;
else
    number = k;
```

```
printf( number);
```



C Programming

```
if (      )
{
}
else {
    }
```

GO if
CLASSES



if ()
[if ()
[if ()]
else

if ()
{ if ()
{ if ()
} else



Switch-case Statement



C Programming

```
switch(x) {  
    case constant1:  
        ...  
        break  
  
    case constant2:  
        ...  
        break  
  
    default:  
        ...  
        break  
}
```

must be an integer



$$2+3 = 5$$

GO
CLASSES
int x;

Char c;

Short s;



Rules for using switch statement

- The expression (after switch keyword) must yield an **integer** value i.e the expression should be an integer or a variable or an expression that evaluates to an integer.
- The case statements and the default statement can occur in any order in the switch statement.
- The case **label** values must be unique *and constant.*
- The case label must end with a colon(:)



C Programming

```
int main() {  
    int num = 8;  
    switch (num) {  
        case 7:  
            printf("Value is 7");  
            break;  
        case 8:  
            printf("Value is 8");  
            break;  
        case 9:  
            printf("Value is 9");  
            break;  
        default:  
            printf("Out of range");  
            break;  
    }  
    return 0;  
}
```

Value is 9





```
int main() {  
    int num = 8;  
    switch (num) {  
        case 7:  
            printf("Value is 7");  
            break;  
        case 8:  
            printf("Value is 8");  
            break;  
        case 9:  
            printf("Value is 9");  
            break;  
        default:  
            printf("Out of range");  
            break;  
    }  
    return 0;  
}
```

Value is 8



C Programming

Question

What will be the output for the program given ?

```
int main() {  
    int num = 2;  
    switch (num+2) {  
  
        printf("Hey");  
  
        case 1:  
            printf("Case1: Value is %d", num);  
        case 2:  
            printf("Case2: Value is %d", num);  
        case 3:  
            printf("Case3: Value is %d", num);  
  
        default:  
            printf("Defualt Value is %d", num);  
    }  
}
```



Question

What will be the output for the program given ?

Default value is 2

```
int main() {  
    int num = 2; 4  
    switch (num+2) {  
        printf("Hey");  
  
        case 1:  
            printf("Case1: Value is %d", num);  
        case 2:  
            printf("Case2: Value is %d", num);  
        case 3:  
            printf("Case3: Value is %d", num);  
  
        default:  
            printf("Defualt Value is %d", num);  
    }  
}
```



C Programming

```
#include <stdio.h>
int main() {
    int num =2;
    switch(num+2){
        printf("hey");
        case 1:
            printf("Case1: Value is %d", num);
        case 2:
            printf("Case2: Value is %d", num);
        case 3:
            printf("Case3: Value is %d", num);
        default:
            printf("Default Value is %d", num);
    }
}
```



Question

What will be the output for the program given ?

```
#include <stdio.h>
```

```
int main() {  
  
    int month = 2;  
    int year = 2000;  
    int numDays = 0;  
  
    switch(month){  
  
        case month:  
            printf("%d", numDays);  
        }  
    }  
}
```



Question

What will be the output for the program given ?

```
#include <stdio.h>
```

```
int main() {
```

```
    int month = 2;  
    int year = 2000;  
    int numDays = 0;
```

```
    switch(month){
```

```
        case month:  
            printf("%d", numDays);  
    }
```

error

variable
is not allowed

}

A screenshot of a Mac OS X desktop environment. On the left, a code editor window titled 'a.c' shows a C program with line numbers from 1 to 17. On the right, a terminal window titled 'Desktop — zsh — 80x24' displays the output of running 'gcc a.c -w', highlighting an error at line 12, column 18.

```
#include <stdio.h>
int main() {
    int month = 2;
    int year = 2000;
    int numDays = 0;

switch(month){
    case month:
        printf("%d", numDays);
}
}
```

```
(base) sachinmittal@Sachins-MacBook-Pro Desktop % gcc a.c -w
a.c:12:18: error: expression is not an integer constant expression
    case month:
                 ^
1 error generated.
(base) sachinmittal@Sachins-MacBook-Pro Desktop %
```

fall - through



```
#include <stdio.h>
int main() {
    int num =3;
    switch(num){
        case 1:
            printf("Case1: Value is %d \n", num);
        case 2:
            printf("Case2: Value is %d \n", num);
        case 3:
            printf("Case3: Value is %d \n", num);
        default:
            printf("Default Value is %d \n ", num);
    }
}
```

1

```
1 #include <stdio.h>
2     int main() {
3
4     int num =3;
5
6     switch(num){
7
8
9     case 1:
10        printf("Case1: \n");
11
12    case 3:
13        printf("Case3: \n");
14
15    case 2:
16        printf("Case2: \n");
17        break;
18
19
20
21    default:
22        printf("Default:");
23 }
```

c a.c

Desktop -- zsh -- 80x24

```
(base) sachinmittal@Sachins-MacBook-Pro Desktop % gcc a.c -w
(base) sachinmittal@Sachins-MacBook-Pro Desktop % ./a.out
```

```
Case3:
```

```
Case2:
```

```
(base) sachinmittal@Sachins-MacBook-Pro Desktop %
```

```
#include <stdio.h>
int main() {
    int num =3;
    switch(num){
        case 1:
            printf("Case1: \n");
        case 3:
            printf("Case3: \n");
        default:
            printf("Default: \n");
        case 2:
            printf("Case2: \n");
            break;
    }
}
```

```
(base) sachinmittal@Sachins-MacBook-Pro Desktop % gcc a.c -w
(base) sachinmittal@Sachins-MacBook-Pro Desktop % ./a.out
Case3:
Default:
Case2:
(base) sachinmittal@Sachins-MacBook-Pro Desktop %
```



GATE 2012 Question

What will be the output
for the program given ?

GATE CSE 2012 | Question: 3



What will be the output of the following C program segment?

35

```
1 char inChar = 'A';
2     switch ( inChar ) {
3         case 'A' : printf ("Choice A \ n");
4         case 'B' :
5         case 'C' : printf ("Choice B");
6         case 'D' :
7         case 'E' :
8             default : printf ("No Choice");
9     }
```



- A. No Choice
- B. Choice A
- C. Choice A
 Choice B No Choice
- D. Program gives no output as it is erroneous

gatecse-2012

programming

easy

programming-in-c



C Programming

```
#include <stdio.h>

int main() {

    char inChar = 'A';
    switch ( inChar ) {
        case 'A' : printf ("Choice A \n");
        case 'B' :
        case 'C' : printf ("Choice B");
        case 'D' :
        case 'E' :
        default : printf ("No Choice");
    }
}
```

Choice A
Choice B No choice



Question

What will be the output for the program given ?


Value

```
#include <stdio.h>

int main() {
    int value = 0;
    switch ( value ) {
        default : value++;
        case 2 : printf ("Humans are human centric ");
                    break;
        case 1 : printf ("This is inhuman ");
    }
    printf("%d", value);
}
```



Question

What will be the output for the program given ?



```
#include <stdio.h>

int main() {
    int value = 0;
    switch ( value ) {
        default : value++;
        case 2 : printf ("Humans are human centric ");
                    break;
        case 1 : printf ("This is inhuman ");
    }
    printf("%d", value);
}
```

Humans are human centric

switch ()

Search for the matching case

if nothing matches then go
to default and fall through till
break



Loops in C Programmes



Suppose that you need to print **Programming is fun!** 100 times.

```
1 print("Programming is fun!")
2 print("Programming is fun!")
3 print("Programming is fun!")
...
98 print("Programming is fun!")
99 print("Programming is fun!")
100 print("Programming is fun!")
```

100
times



Loops





while loop

```
while (condition test) {  
    //Statements to be executed repeatedly  
}
```



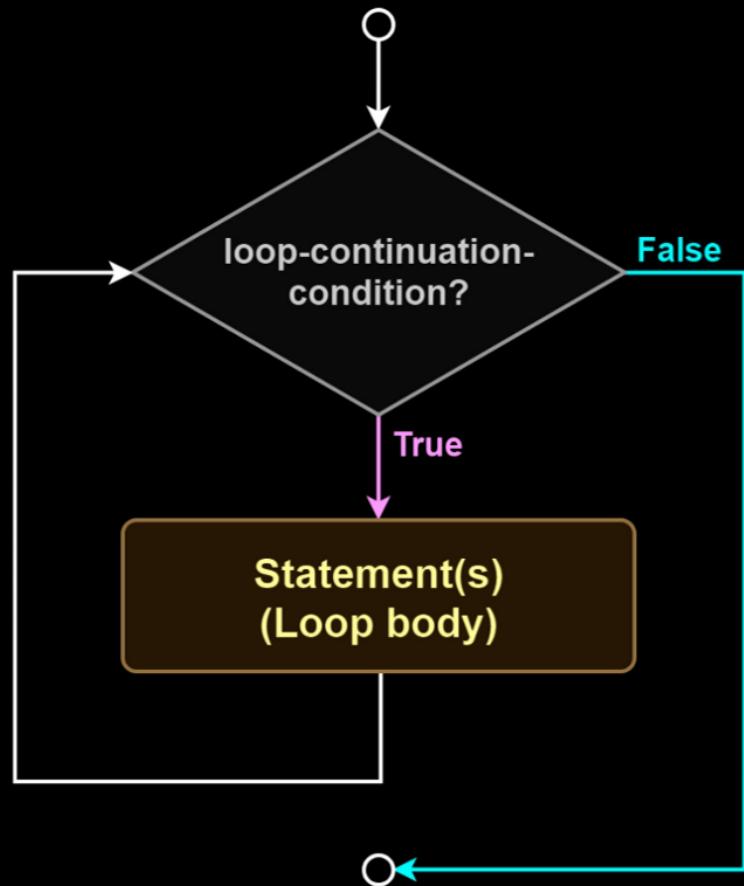
C Programming

```
int count = 0;  
while (count < 100)  
{  
    printf("Programming is fun!\n");  
    count = count + 1;  
}
```

Count
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100



C Programming



SSES



C Programming

```
int weight; // 75
while ( weight > 65 ) {
    printf ("Go, exercise, ");
    printf ("then come back. \n");
    printf ("Enter your weight: ");
    scanf ("%d", &weight);
}
```

TO 63



Question

What is the value of x after the following C code snippet is executed?

```
int x = 10;  
while (x > 0) {  
    x = x - 3;  
}  
'
```



A. -1

B. -2

C. -3

D. -4



Question

What is the value of x after the following C code snippet is executed?

```
int x = 10;  
while (x > 0) {  
    x = x - 3;  
}  
'
```



- A. -1
- B. 2
- C. -3
- D. -4

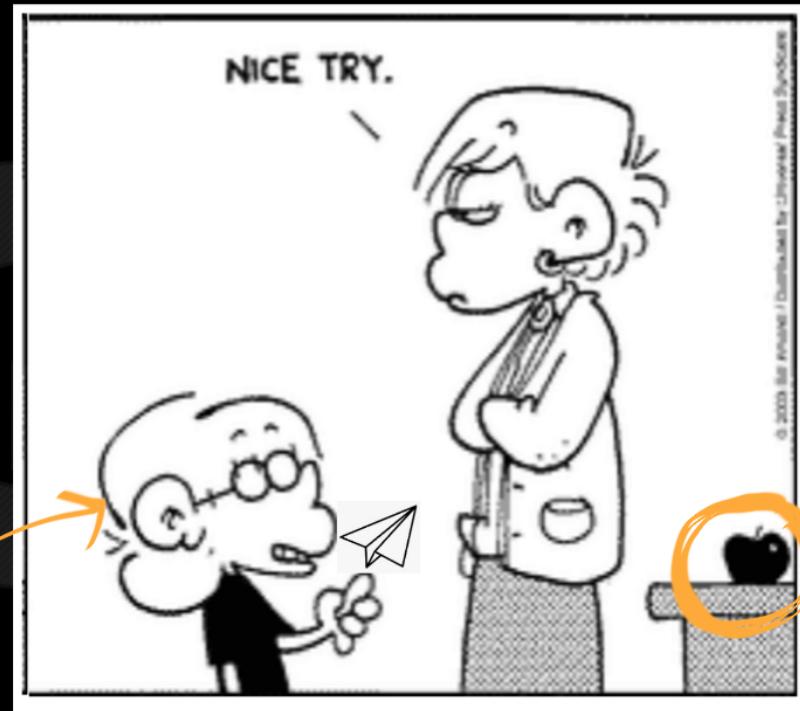


for Loop

```
for (initialization; test; update)
{
    // statements inside the body of loop
}
```



C Programming



© 2003 Bill Amend / Dilbert distributed by Universal Press Syndicate



C Programming

```
#include <stdio.h>
int main(void)
{
    int count;
    for(count=1; count<=500; count++)
        printf("I will not throw paper airplanes in class.");
    return 0;
}
```

MEND 10-3





C Programming

```
int P;  
for (p = 10; p > 0; p--)  
    printf("%d", p);
```

\rightarrow
 $p=10$

$\cancel{10}$ $\cancel{9} \cancel{8}$
 P

$\boxed{10 \cancel{9} \cancel{8} \cancel{7} \cancel{6} \cancel{5} \cancel{4} \cancel{3} \cancel{2} \cancel{1}}$



for has three parts, *each of which is optional.*

```
for (initialization; test; update)
{
    // statements inside the body of loop
}
```



C Programming

```
#include<stdio.h>
void main()
{
    int i=0;
    for(;i<=9;)
    {
        i++;
        printf("\n value is:%d",i);
    }
}
```

value is 1
value is 2 3

4
value is 5
value is 6 7 8 9 10



Question

```
#include<stdio.h>
int main()
{
    int i;
    for(i = 0;i<=3;i++);
    printf("%d", i);
    return 0;
}
```

for () ;
≡ for () {
 }

- A. Compilation error
- B. 1 2 3
- C. 4
- D. 0 1 2 3



C Programming

```
for(i = 0;i<=3;i++);  
printf("%d", i);
```

```
for(i = 0;i<=3;i++)  
{  
    printf("%d", i);  
}
```





do - while Loop



do - while Loop

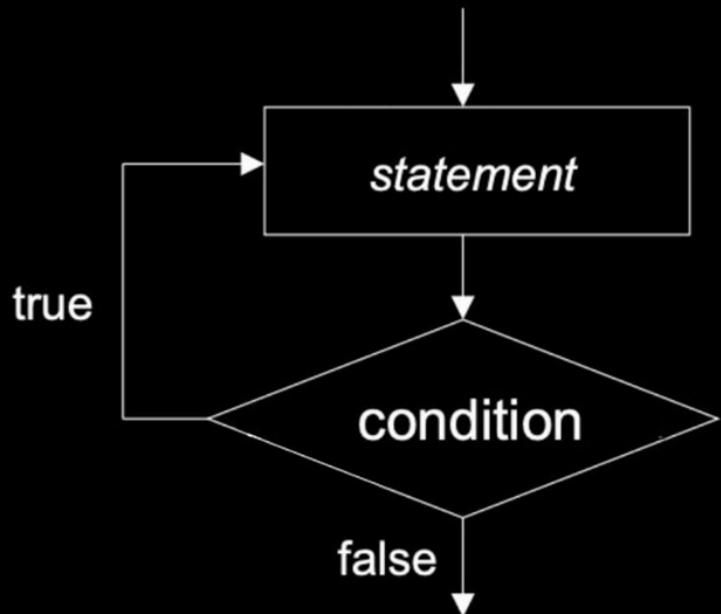


do while Loop

```
do
{
    // statements
}
while(condition);
```



GC
CL

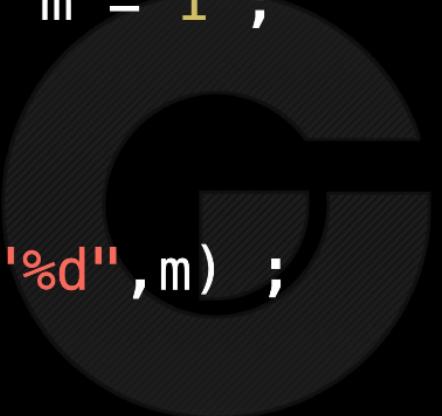




C Programming

```
#include<stdio.h>
main()
{
    int n = 0, m = 1 ;
    do
    {
        printf("%d",m) ;
        m++ ;
    }
    while (m <= n) ;
}
```




GO
CLASSES





```
#include<stdio.h>
main()
{
    int n = 4, m = 1 ;
    do
    {
        printf("%d",m) ;
        m++ ;
    }
    while (m <= n) ;
}
```

~~n~~ 0

~~m~~ 1 2 3 4

GO
CLASSES

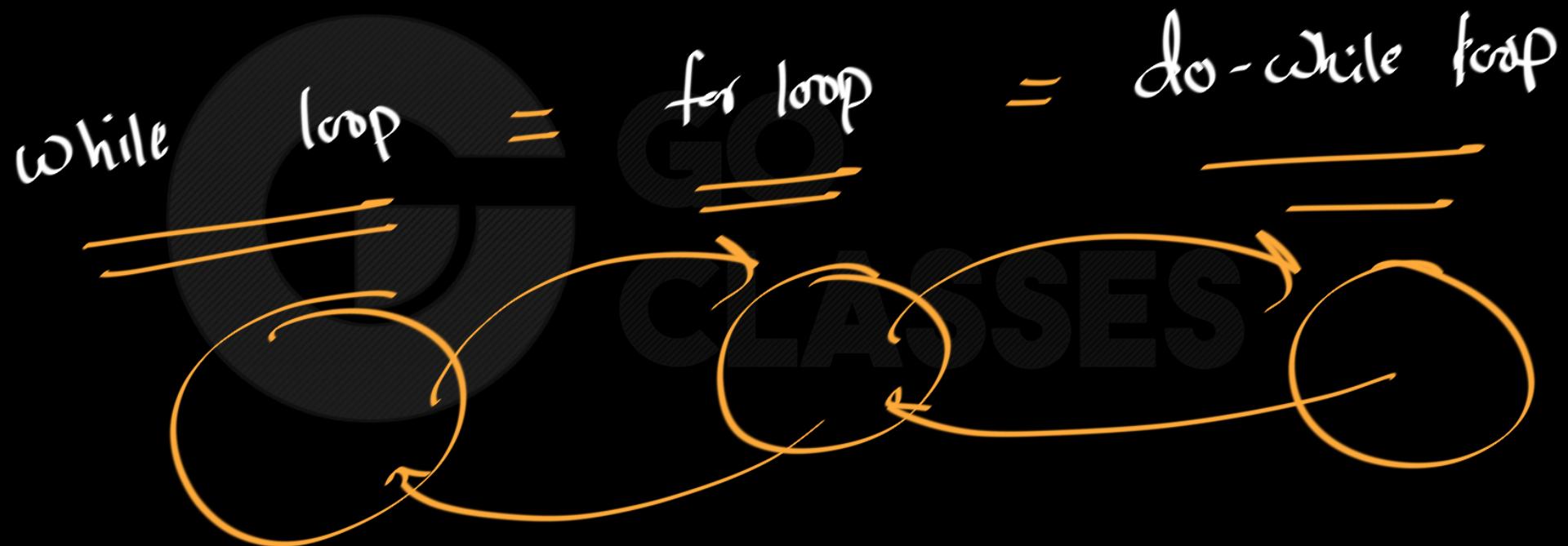
5 6 7 8



```
#include<stdio.h>
main()
{
    int m=3;
    do
    {
        printf ("%d\n",m);
        m=0;
    } while(m>0);
}
```

 3
m

 3
output





Break Statement



C Programming

```
#include <stdio.h>
int main ()
{
    int a;
    while (1)
    {
        printf("enter the number:");
        scanf("%d", &a);
        if ( a == 0 )
            break;
    }
    return 0;
}
```

~~a~~ 3 0

S
→ come out of the innermost loop
(only one loop)



I wrote a while(true) loop



With break
statement right?



I wrote a while(true) loop

With break
statement right?



C Programming

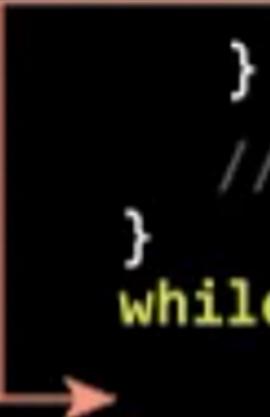
```
while (testExpression) {  
    // codes  
    if (condition to break) {  
        break;  
    }  
    // codes  
}
```

← Closing of While

S



```
do {  
    // codes  
    if (condition to break) {  
        break;  
    }  
    // codes  
}  
while (testExpression);
```





```
for (init; testExpression; update) {  
    // codes  
    if (condition to break) {  
        break;  
    }  
    // codes  
}
```





C Programming

```
while (testExpression) {  
    // codes  
    if (condition to break) {  
        break;  
    }  
    // codes  
}
```

```
do {  
    // codes  
    if (condition to break) {  
        break;  
    }  
    // codes  
}  
while (testExpression);
```

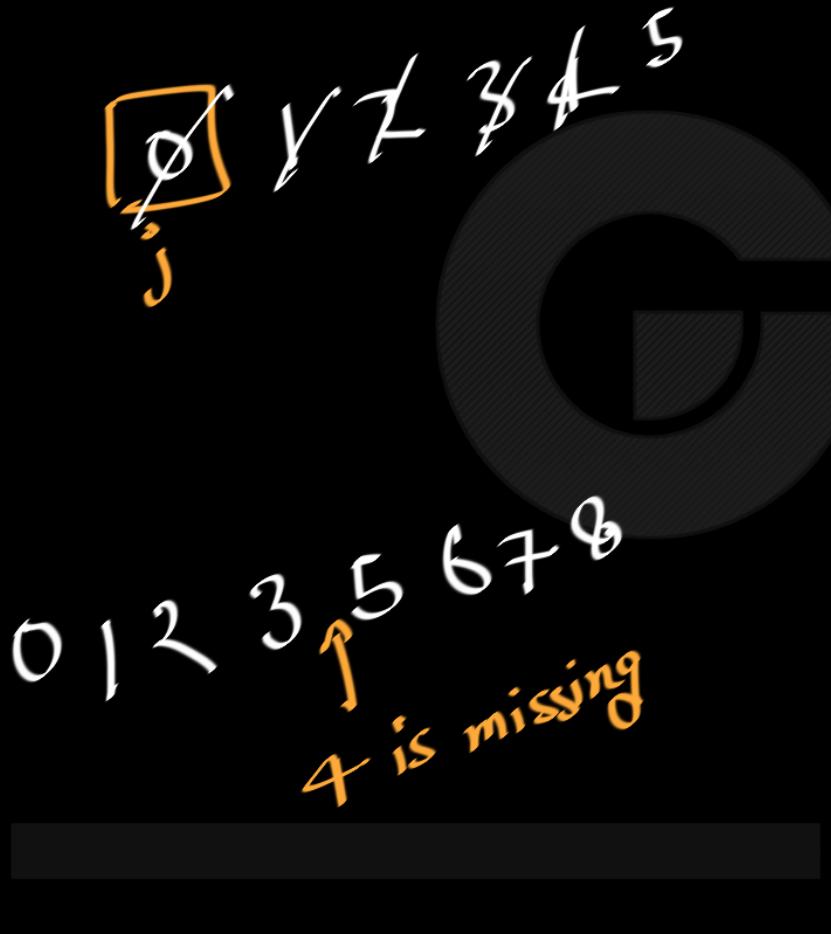
```
for (init; testExpression; update) {  
    // codes  
    if (condition to break) {  
        break;  
    }  
    // codes  
}
```

```
    }  
    // codes  
}
```



Continue Statement





```
#include <stdio.h>
int main()
{
    for (int j=0; j<=8; j++)
    {
        if (j==4)
        {
            /* The continue statement is encountered when
             * the value of j is equal to 4.
            */
            continue;
        }

        /* This print statement would not execute for the
         * loop iteration where j ==4 because in that case
         * this statement would be skipped.
        */
        printf("%d ", j);
    }
    return 0;
}
```

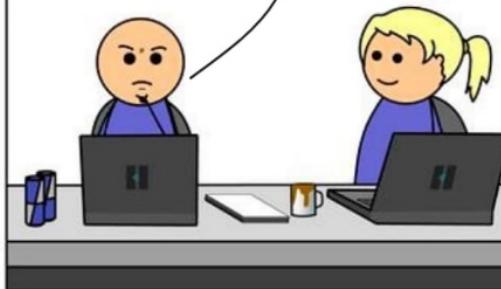
```
→ while (testExpression) {  
    // codes  
    if (testExpression) {  
        → continue;  
    }  
    // codes  
}
```

```
do {  
    // codes  
    if (testExpression) {  
        → continue;  
    }  
    // codes  
}  
→ while (testExpression);
```

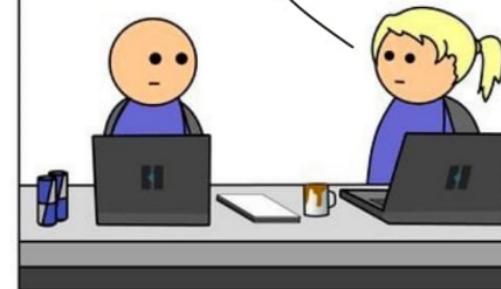
```
→ for (init; testExpression; update) {  
    // codes  
    if (testExpression) {  
        → continue;  
    }  
    // codes  
}
```

C Programming

I want to tell you a joke
about continue statement



Ok, Tell me



But, I decide to skip it





Output ?

i = 0 1 2 - - \$

Output: 0 1 2 3 4

```
#include <stdio.h>

int main(void){
    int hours = 10;
    int i=0;

    for(i=0;i<hours;i++){
        if(i==5){
            break;
        }

        printf("%d ",i);
    }

    return 0;
}
```



Output ?

0 1 2 3 4 5 6 7 8 9
|
5 is missing

```
#include <stdio.h>

int main(void){
    int hours = 10;
    int i=0;

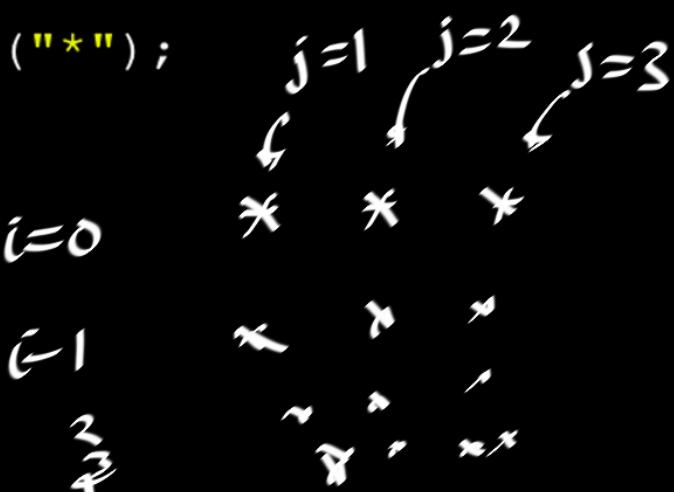
    for(i=0;i<hours;i++){
        if(i==5){
            continue;
        }
        printf("%d ",i);
    }
    return 0;
}
```

Output ?

i=0

j=1,2,3,4...

```
int main() {  
    // nested for loops with break statement  
    // at inner loop  
    for (int i = 0; i < 5; i++) {  
        for (int j = 1; j <= 10; j++) {  
            if (j > 3)  
                break;  
            else  
                printf("*");  
        }  
        printf("\n");  
    }  
    return 0;  
}
```





C Programming

Output ?

```
int main() {  
    // loop initialization expression  
    int i = 1;  
  
    // infinite while loop  
    while (1) {  
        if (i > 10)  
            break;  
  
        printf("%d ", i);  
        i++;  
    }  
  
    return 0;  
}
```

1 2 3 4...9 10