

```
1. int main ()  
{  
    int i = 16;  
    i = ! i > 15;  
    printf ("%d", i);  
    return 0;  
}
```

Output = 0
 $i = 0 > 14 \Rightarrow i = 0$
(False)

```
2. int main ()  
{  
    int a = 5;  
    a = 1, 2, 3;  
    printf ("%d", a);  
    return 0;  
}
```

Output :
1

```
3. int main ()  
{  
    int x = 2;  
    (x & 1) ? printf ("true") : printf ("false");  
    return 0;  
}
```

Output : False
When multiplied
returns 0

4. int main ()

```
int i;  
if (true)  
    printf ("A");  
else  
    printf ("B");  
return 0;
```

Output:

Lexer

undefined
symbol true

5. int main ()

```
{  
static int i;  
for(i++; ++i; i++)  
{  
    printf ("%d", i);  
    if (i == 6)  
        break;  
}  
return 0;
```

Output

2,4,6

1. void main()

```
{  
    int x = 5;  
    if (x < 1);  
    printf ("A");  
}
```

Output: A

2. int main()

```
{  
    int x;  
    for (x = -1; x <= 10; x++)  
    {  
        if (x < 5)  
            continue;  
        else  
            break;  
        printf ("A");  
    }  
    return 0;  
}
```

Output:
Nothing will
print.

3. int main()

```
{  
    int a = 300, b, c;
```

```
if (a >= 400)
    b = 300;
    c = 200;
    printf ("%d %d", b, c);
    return 0;
}
```

Output :
Garbage value, 200

4. void main()

```
{ int i=0;
for(;i;)
    printf ("A");
}
```

Output :
points nothing

5. int main()

```
{ int x=4;
while (x == 1)
{ x=x-1;
    printf ("%d", x);
    x--;
}
}
```

Output :
points nothing

6. `int main()`

```
{  
    if (sizeof('0'))  
        printf("A");  
    else  
        printf("B");  
    return 0;  
}
```

Output:
A

7. `int main()`

```
{  
    char ch = 0;  
    while (ch == '0')  
    {  
        printf("A");  
        break;  
    }  
    return 0;  
}
```

Output:
Prints Nothing

8. `int main()`

```
{  
    int i = 0;  
}
```

while (++i)

{
 i = --i ? i = 0 : i = 1; }

printf ("%d", i);
return 0;

Output:
Infinite Loop

9x10ms (9ms)

2 sec - 100
200m/s 100m
1000m/s 100
Tubular 0.5
0.5 + 9 x 10 x 0.0
0.5 + 9 x 10 x 0.0
100 m/s 10
0.1sec 2 m/s
2 x 10^8 m/s
- 10 sec

1. int main()

```
int i;  
if (faintf ("0"))  
    i = 3;  
else  
    i = 5;  
faintf ("%d", i);  
return 0;
```

Output : 03

2. int i;

int main()

```
{  
    if (i);  
else  
    faintf ("Else");  
return 0;
```

Output : else

block is
executed

i is globally defined
& its default
value is zero.

3. int main()

```
{  
    int c = 5, no = 10;  
    do {  
        no /= c;
```

```
{ while (c--);  
    printf ("%d", no);  
    return 0;  
}
```

Output :
ERROR

7. Void main ()
{
 int a = 10, b = 2, x = 0; Output = 80
 x = a + b * a + 10 / 2 * a;
 printf ("Value is = %d", x);
}

8. int main ()
{
 int i = 3; Output = ~~3~~
 printf ("%d", (++i)++); Compile time
 return 0; error
}

9. Void main () {
 int a = 5, b = 10, c = 1 ; Output =
 if (a && b > c) {
 printf ("____"); Compilation
 }
 else {
 break ; Error
 }
}

```
10. Void main () {  
    if (sizeof (void))  
        printf (" A");  
    else  
        printf (" B")  
}
```

Output = compil-
ation Error

```
11. Void main () {  
    int m = 5, n = 10, q = 20;  
    if (q / n * m)  
        printf (" A");  
    else  
        printf (" B");  
    printf (" C");  
}
```

Output = A C

```
12. Void main () {  
    int a = 5, b = 10;  
    if (++a || ++b)  
        printf ("%d %d", a, b);  
    else  
        printf (" C");  
}
```

Output → 6
10

21. int main()

{

 int i = 0;
 switch(i){
 case '0': printf("A"); break;
 case '1': printf("B"); break;
 default: printf("C");
 } return 0;

 } Output = C ④

characters → ASCII values
 {
 (0 → 48)
 (1 → 49)

22. int main()

{
 int i;
 for (i = 1; i != 10; i += 2)
 printf("A");
 return 0;
}

Output = Infinite
Loop
i will never
be equal to 10

23. int main()

{
 int i; goto loop;
 for (i = 0; i < 10; i++)
 printf("A");
 loop:
 break;
 return 0;
}

Output = No
output

```
18. int main()
{
    int a;
    a = sizeof(.5.6);
    printf("%d", a);
    return 0;
}
```

Output = 2
GCC = 4

```
19. int main()
{
    int i = 1, j = 1;
    for (--i && j++; i < 10; i += 2)
    {
        printf("A");
    }
    return 0;
}
```

Output = 5 times
A

```
20. int main()
{
    for (5; 2; 2)
    printf("A");
    return 0;
}
```

Output = Infinite
Loop

Condition place of
loop → non-zero
value

4. int main() {
 printf ("%d %d %d", sizeof(3.14),
 sizeof(3.14f), sizeof(3.14L));
 return 0; }
Output → 8 4 10

Double
→ const.
long
double
Conet.

Floating
pt. Constant

5. int main () {
 int i=5;
 int a = ++i + ++i + ++i;
 printf ("%d", a);
 return 0; }
Output → 6 + 7 + 8 = 21

6. Void main () {
 printf (" Value is = %d", (10++)); }
Output → Error

(Operator works on Variables only).

①

```
1. #include <stdio.h>
int main() {
    int i = 1;
    i = 2 + 2 * i++;
    printf ("%d", i);
    return 0;
}
```

Output = 5

$$\begin{aligned} i &= 2 + 2 * 1 \\ &= 4 \\ i &= 4 + 1 = 5 \end{aligned}$$

```
2. void main () {
    int x ;
    x = 10, 20, 30;
    printf ("%d", x);
    return 0;
}
```

Output = 10

assignment operator
has more precedence
than comma operator

```
3. int main () {
    int a = 0, b = 10;
    if (a == 0) {
        printf ("true");
    } else {
        printf ("false");
    }
    return 0;
}
```

Output = false

0 → false

Non-zero → true

if (0) → cond.
is false

11. void main()

```
{  
    const char var = 'A';  
    ++var;  
    printf("%c", var);  
}
```

Output : Error
Cannot change
the value of
constant.

12. void main()

```
{  
    int x = (20 || 40) && (10);  
    printf("%d", x);  
}
```

Output = 1
 $(20 \text{ || } 40) \rightarrow \text{Both}$
are non-zero,
so it will return 1
 $1 \text{ && } 10 \rightarrow \text{both}$
are non-zero so return
1

13. int main()

```
{  
    int i, j;  
    i = j = 2, 3;  
    while (--i && j++)  
        printf("%d %d", i, j);  
    return 0;  
}
```

Output = 1 & 3
 $i = 2, j = 2$
 $(1 \& 2) \rightarrow \text{True}$
 $i = 1 \& j = 3$
 $i = 0, j = 3 (0 \& 3)$
while cond. is
terminated False

14. int main() {

```
    int i = 1;
```

for ($i = 0$; $i = -1$; $i = 1$)
{
 printf ("%d", i);
 if ($i \neq 1$) break;
}

Output = -1

$i = -1 \rightarrow$ non-zero
so cond. is true

25. int main () {
 int goto = 5;
 printf ("%d", goto);
 return 0;
}

Output = Comp. compilation error
(goto is a keyword)

26. int main () {
 int a = 2, b = 7, c = 10;
 c = a == b;
 printf ("%d", c);
 return 0;
}

Output = 0

a == b returns
2 values
0 if false
1 " true

27. int main () {
 int x = 100, y = 20, z = 5;
 printf ("%d %d %d");
 return 0;
}

Output = 5 20
100

Garbage value
(GCT)

Stack \rightarrow LIFO