

Practical task

Page No.	
Date	

1 what will be the output of given code ?

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

```
#include <conio.h>
```

```
void main ()
```

```
{
```

```
int i=0,j=1,k=2,m;
```

```
m = i++ || j++ || k++
```

```
printf ("%d %d %d %d",m,i,j,k);
```

```
}
```

output

1 1 2 2

Description :-

If we think above code we have statement ~~int a=5, b~~ $i=0, j=1, k=2, m;$,

$m = i++ || j++ || k++;$

Here we use two operators

$++ \quad ||$

Here $||$ (logical or) have first priority

$++$ low priority

$||$ get executed first & $++$ then.

$m = 0 \quad || \quad 1 \quad || \quad 1$

$\begin{matrix} f & \sqcup & t \\ & \sqcap & \\ t & \sqcup & 1 \end{matrix}$

$\therefore m = 1$

Now second priority + from table

$m = \perp \quad 1 \quad | \quad 2 \quad | \quad 2$ (doubtful)

2 #include <stdio.h>
 #include <conio.h>
 void main ()
 {
 int c = --2 ;
 printf ("c = %d", c);
 }

Compile time error -

late cannot assign increment /
 decrement operator to constant value

③ #include <stdio.h>
 #include <conio.h>
 int main () {
 int i = 3;
 int l = i / -2;
 int k = i % -2;
 printf ("%d %d %d", l, k);
 return 0;
 }

output : -11

Description

```
4 #include <stdio.h>
int main()
{
    int i=5
    i = i/3;
    printf ("%d\n", i);
    return 0;
}
```

output : -1

Description

in above code
 5 constant value assigned to the
 integer variable 5
 & then 5/3 assigned to i variable
 so the quotient is 1

```
5 #include<stdio.h>
int main()
{
    int i=-5;
    i = i / 3
    printf ("%d", i);
    return 0;
}
```

output - 1

```
6 #include <stdio.h>
void main()
{
    int a=5, b=-7, c=0, d;
    d = ++a && ++b || ++c;
    printf ("\n %d %d %d %d", a, b, c, d);
}
```

output

6 -6 0 1

Description

Here in above equation

operator = ++a && ++b || ++c;

first priority to ++ increment
(pre increment)

$d = ++a \quad f = ++b \quad l = ++c$
 $\overbrace{+}^6 \quad \overbrace{+}^6 \quad \overbrace{-}^6$
 true (\pm) } second condition
 because first
 condition is true.

In logical $l l$ If first condition is true then compiler skip the second condition.

```

7 #include<stdio.h>
void main()
{
  int a = -5;
  int k = (a++, ++a);
  printf ("%d\n", k);
}
output -3
  
```

Description.

Here in above equation first priority to $++$ pre increment operation $\therefore a = -4$ & then second priority to post increment

8 #include <stdio.h>
int main ()
{
int x = 2;
x = x << 1;
printf ("%d\n", x);
}

B.V 0010
LS \rightarrow []
2
0100
4

output = 4

Binary conversion

1010110
left shift
1011100

9 #include <stdlib.h>
#include <conio.h>
int main ()
{
int x = -2;
x = x >> 1;
printf ("%d\n", x);
}

B.V 0010
RS ~~1101~~
-3 0110
0011

output : -1

Description

```
10) #include <stdio.h>
int main() {
    if (~0 == 1)
        printf("Yes\n");
    else
        printf("No\n");
}
```

Output : No

Description :

11) #include <cslib.h>
int main()
{
 int y=0;
 If (1 | (y=1))
 printf ("y is %d \n", y);
 else
 printf ("%d \n", y);
}

output 1

Description:

Here $y = 1$ the value reassigned to the y variable.

$1 | 1 = 1$ (bitwise operator)

∴ If condition will print 1

12) #include <cslib.h>
#include <conio.h>
void main()
{
 int y=1;
 If (y & (y=z))
 printf ("true %d \n", y);
 else
 printf ("false %d \n", y);
}

output

y true 2

Description

Here y=1:

& in if condition 0010

2 value assigned to the variable
y

$\therefore y \& y = 2$

i.e. 1 & 2

1/true 1/true

1 i.e. true.

Q) #include <stdio.h>

int main()

{ unsigned int res

res = (64 >> (2+1-2)) & (~(1 << 2));

print ("%d\n", res);

return 0;

}

output 82

Description

first priority,

(64 >> (2+1-2))

$\therefore (64 >> 1)$

82

Second priority

6432168 < 2 = 1

0010000000000000

0010000000000000

$\hookrightarrow >> 1$

Now the value

is 82

(~(1 << 2))

0001

0100

④

= ~4 = -5

1010

1010

1011

421

0010000000000000 32

11111011 32

0010000000000000 32

32

```

14 #include <stdio.h>
#include <conio.h>
int main() {
    int i=4, j=8;
    printf("%d,%d,%d\n", i&j, i|j, i&&j);
    return 0;
}

```

output - my op 12 12 true 12

Description

i = 4 , j = 8

i 0100

j 1000

1st

$$\begin{array}{r} 0100 \\ + 1000 \\ \hline 1100 \end{array}$$

2nd

$$\begin{array}{r} 1000 \\ \& 1000 \\ \hline 1000 \end{array}$$

true = 1

$$i \wedge j = 0100$$

$$\begin{array}{r} 0100 \\ + 1000 \\ \hline 1100 \end{array}$$

true = 1

15

```
#include <stdio.h>
#include <conio.h>
int main()
{
    printf ("%ox\n", -1>>1);
    return 0;
}
```

output fff ffffff

Description

(8>>1) = (at x0⁰)

16 #include <stdio.h>
int main()
{
 unsigned int m=32;
 printf ("%x\n", ~m);
 return 0;
}
output ffff.ffff

17] #include <stdio.h>
#include <conio.h>
int main()
{
 printf ("%x\n", -1<<3);
 return 0;
}
output.

18] #include <stdio.h>
#include <conio.h>
int main()
{
printf ("%d>>%d %d>>%d \n", 4>>1, 8>>