

21/11/2025

1) Read two integer values perform bitwise operations.

$a = 4, b = 2$

$4/2$

$4 \& 2$

$4 \wedge 2$

$4 \sim 4$ is

\Rightarrow #include <stdio.h>

int main()

{

int a, b;

printf("Enter two integer values: ");

scanf("%d %d", &a, &b);

printf("\n (a & b) = %d", a & b);

printf("\n (a | b) = %d", a | b);

printf("\n (a ^ b) = %d", a ^ b);

printf("\n (~a) = %d", ~a);

printf("\n (~b) = %d", ~b);

return 0;

}

Output

Enter two integer values: 4, 2

And (a & b) = 0

OR (a | b) = 71441108

XOR (a ^ b) = 71441108

NOT (~a) = -5

NOT (~b) = -71441105

2) $a=4, b=2$

$4 < 2$

$4 > 2$

$4 <= 2$

$4 >= 2$

$4 != 2$

⇒ #include <stdio.h>

int main()

{

int a, b;

printf("Enter two integer values: ");

scanf("%d %d", &a, &b);

printf("\n Results of relational operations: \n");

printf("a < b: %d \n", a < b);

printf("a > b: %d \n", a > b);

printf("a <= b: %d \n", a <= b);

printf("a >= b: %d \n", a >= b);

printf("a != b: %d \n", a != b);

return 0;

}

Output:-

Enter two integer values: 4, 2

Results of relational operations:

a < b: 0

a > b: 1

a <= b: 0

a >= b: 1

a != b: 1

MCQ → increment, decrement, shift operator.

1) What will be the output of the following C code?

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

```
int main()
```

```
{
```

```
int i = 0;
```

```
int x = i++, y = ++i;
```

```
printf("%d %d\n", x, y);
```

```
return 0;
```

a) 0, 2

b) 0, 1

c) 1, 2

d) undefined

⇒ i++ → post-increment

i = 0 + 1

x = 1

++i → pre-increment

i = 0

y = 2

2) What will be the output of the following code?

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

```
int main()
```

```
{
```

```
int i = 10;
```

```
int *p = &i;
```

```
printf("%d\n", *p++);
```

a) 10

b) 11

c) Garbage value

d) Address of i

3) what will be the output of the following code?

```
#include <stdio.h>
void main()
{
    int x = 97;
    int y = size of (x++);
    printf("x is %d", x);
}
```

- a) x is 97 b) x is 98 c) x is 99 d) run time error
- ⇒ x++ no effect

4) what will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    int x = 4, y, z;
    y = --x;
    z = x--;
    printf("%d %d %d", x, y, z);
}
```

- a) 3 2 3 b) 2 3 3 c) 3 2 2 d) 2 3 4

x = 4.

--x (pre-decrement decreases) -

x- (post decrement decreases)

2 3 3

5) #include <stdio.h>

void main()

{

int x = 4;

int *p = &x;

int *k = p++;

int r = p - k;

printf("%d", r);

}

- a) 4 b) 8 c) 1 d) run time error

9) #include <stdio.h>

int main()

{

if (1 & 8)

printf ("Honesty");

if ((1 & 0x000f) == 8)

printf ("is the best policy\n");

}

a) Honesty is the best policy

d) No output

b) Honesty is the best policy

10) #include <stdio.h>

int main()

{

int a = 2;

if (a >> 1)

printf ("%d\n", a);

}

a) 0 b) 1 c) 2 d) No output

$a \gg y$ is right shift operator

$a \gg 1$, $2 \gg 1$ gives 1 (non zero value)

11) #include <stdio.h>

int main()

{

int i, n, a = 4;

scanf ("%d", &n);

for (i = 0; i < n; i++)

a = a * 2;

}

a) logical shift left b) logical shift right

c) Arithmetic shift right d) Bitwise exclusive OR

12) #include <stdio.h>
 int main()
 {
 unsigned int a = 10;
 a = ~a;
 printf("%i.d\n", a);
 }
 a) -9 b) -10 c) -11 d) 10

$\Rightarrow \sim a = -(a+1) = -11$

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

a) 4 b) 1 c) Depend on the compiler d) depends on the endianness of the machine.

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

a) 1 b) -1 c) $2^{31}-1$ considering into be 4 bytes d) either -1 or 1

15) #include <stdio.h>
 int main()
 {
 if (no == 1)
 printf("yes\n");
 else
 printf("no\n");
 }

a) yes b) no c) compile time error d) undefined.

```

19) #include <stdio.h>
void main()
{
    int k=8;
    int x=0 == 1 & k++;
    printf ("%d.%d\n", x, k);
}

```

a) 0.9 b) 0.8 c) 1.8 d) 1.9

⇒ 0.8 == 1 & k++

```

20) #include <stdio.h>
void main()
{
    char a = 'a';
    int x = (a % 10) ++;
    printf ("%d\n", x);
}

```

a) 6 b) junk value c) compile time error d) 7

```

21) #include <stdio.h>
void main()
{
    1 < 2 ? return 1 : return 2;
}

```

a) returns 1 b) returns 2 c) varies d) compile time error

```

22) #include <stdio.h>
void main()
{
    unsigned int x = -5;
    printf ("%d", x);
}

```

a) run time error b) tries c) -5 d) 5

```

23) #include <stdio.h>
int main()
{
    int x = 2, y = 2;
    x /= x / y;
    printf ("%d\n", x);
    return 0;
}

```

a) 2 b) 1 c) 0.5 d) undefined behaviour.

```

24) #include <stdio.h>
int main()
{
    int x=1, y=0;
    x&&=y;
    printf("%d\n", x);
}

```

☒ a) compile time error b) 1 c) 0

d) undefined behaviour

25) what will be the value of the following expression
 "(x=foo()) != 1 considering foo() returns 2"

☒ a) 2 b) true c) 1 d) 0

26) operation "a = a * b + a" can also be written as

☒ a) a* = b+1; b) (c = a*b, a = c+a) != a; c) a = (b+1)*a
 d) All of the mentioned.

27) what will be the final value of c in the following C statement? (initial value : c = 2)

c < c = 1;

a) c = 1; b) c = 2; c) c = 3; ☒ d) c = 4;

28) #include <stdio.h>

int main()

{

int a = 1, b = 2

a += b -= a;

printf("%d %d", a, b);

}

a) 1 1 b) 1 2 ☒ c) 2 1 d) 2 2

⇒ b -= a ⇒ b 2 - 1 = 1

a += (b) ⇒ a 1 + 1 = 2

3) $a=4, b=2$
 $a==b$

\Rightarrow #include <stdio.h>

int main()

{

int a, b;

printf("Enter two integer values:");

scanf("%d %d", &a, &b);

if (a == b) {

printf("Both numbers are equal.\n");

} else {

printf("Both number are not equal.\n");

return 0;

}

Output:

Enter two integer values: 4, 2

Both numbers are not equal.

```

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);
}

```

a) 6, -600 b) 6-5 0 1 c) -6-6 0 1 d) 6-6 0 1

$$++a = 5+1 = 6$$

$$++b = -7+1 = -6$$

$$++c = 0 \quad \text{--- (1) true}$$

$$d) ++a \&\& ++b - \text{true} - 1$$

```

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

```

a) -4 b) -5 c) 4 d) -3

$$a++ = -5 \quad | \quad ++a = -4$$

```

8) #include <stdio.h>
int main()
{
    int c=2^3;
    printf("%d\n", c);
}

```

a) 1 b) 8 c) 9 d) 0

$$0 \ 0 = 0$$

$$1 \ 1 = 0$$

$$1 \ 0 = 1$$

$$0 \ 1 = 1$$

$$2 = 0010$$

$$3 = 0011$$

16) #include <stdio.h>

int main()

{

int y=0;

if (1 || (y=1))

printf("y is %d\n", y);

else

printf("%d\n", y);

✓ a) y is 1 b) 1 c) run time error d) undefined

1 || 1 = 1

1 || (y=1)

17) #include <stdio.h>

int main()

{

int y=1;

if (y & (y=2))

printf("true %d\n", y);

else

printf("false %d\n", y);

}

✓ a) true 2 b) false 2 c) either true or false 2 d) true 1

y & (y=2)

1 & 2 = 0

18) #include <stdio.h>

void main()

{

int x=0;

if (x=0)

printf("Its zero\n");

else

printf("Its not zero\n");

}

✓ a) Its not zero b) Its zero c) Run time error d) None

29) #include <stdio.h>

int main()

{

int a=4, n, i, result=0;

scanf("%d", &n);

for (i=0; i<n; i++)

result+=a;

}

a) Addition of a and n b) Subtraction of a and n

~~c) Multiplication of a and n~~ c) division of a and n

30) which of the following is an invalid assignment operator?

a) a*=10; b) a/=10; c) a|=10; ~~d) None of mentioned~~