



Homework 3





All operators with their associativity and precedence

Operator	Description	Associativity	
() [] . -> ++ --	Parentheses or function call Brackets or array subscript Dot or Member selection operator Arrow operator Postfix increment/decrement	left to right	
++ -- + - ! ~ (type) * & sizeof	Prefix increment/decrement Unary plus and minus not operator and bitwise complement type cast Indirection or dereference operator Address of operator Determine size in bytes	right to left	



* / %	Multiplication, division and modulus	left to right
+ -	Addition and subtraction	left to right
<< >>	Bitwise left shift and right shift	left to right
< <=	relational less than/less than equal to	left to right
> >=	relational greater than/greater than or equal to	
== !=	Relational equal to or not equal to	left to right
&	Bitwise AND	left to right
^	Bitwise exclusive OR	left to right
	Bitwise inclusive OR	left to right
&&	Logical AND	left to right
	Logical OR	left to right
? :	Ternary operator	right to left



= += -= *= /= %=&= ^= = <<=>>=	Assignment operator Addition/subtraction assignment Multiplication/division assignment Modulus and bitwise assignment Bitwise exclusive/inclusive OR assignment Bitwise left shift/right shift assignment	right to left
,	comma operator	left to right



Highest and 2nd highest precedence operators



1	OPERATORS	ASSOCIATIVITY
2	() [] -> . ++ -- (postfix) sizeof & * + - ~ ! typecasts ++ -- (prefix)	left to right right to left



Manish Vijay to Everyone 8:33 PM

MV

chatgpt se pucha tha mne

Fruity Penguins Eat Apples, Sometimes
Grapes (for basic operators in C)

F - Function call

P - Postfix increment/decrement

E - Exponentiation (Right to Left)

A - Addition, Subtraction

M - Multiplication, Division, Modulo

G - Greater than, Less than, Greater

than or equal to, Less than or equal to

S - Shift operators (left shift, right
shift)

E - Equality operators (equal to, not
equal to)

A - Bitwise AND

S - Bitwise OR

G - Bitwise XOR

N - Logical NOT

S - Logical AND

G - Logical OR

1 New Message



Question 1

```
#include<stdio.h>
void main()
{
    int i=2, j=3, k=0;
    int p;
    p=(i,k,j);
    printf("%d\n",p);
}
```

↳ 3



Question 2

```
#include<stdio.h>  
  
void main()  
{  
    int ii = 10;  
    ii <<= 1;  
    printf("%d\n",ii);  
}
```



GO
CLASSES

↳ 20



Question 3

```
main()
{
    int var1=1,var2=12,var3=12;
    var1=var2==var3;
    printf("%d", var1);
}
```

L, I

“==” has higher precedence than “=”



Question 4

```
#include<stdio.h>
main()
{
    int var = - -3;    = 3
    printf("var=%d",var);
}
```

→ 3



Question 5

```
#include<stdio.h>
main()
{
    int i = 15, j = 4, m, n;
    m = i > 9; m = 1
    n = j > 2 && j != 2; n = 1
    printf("m = %d n = %d", m, n);
}
```

m = 1 n = 1
true true b, b,

Use the code segment below for problems 6 – 9

GO Classes

```
int x = 7;
int y = 10;
int z = 5;
int result = 0;

result = ++y - 10 || z - 5 && x++;
result += y++ - 11 || z++ - 5 && x++;
result += y + 1 > 11 && (z++ >= 6 || x++);

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

6. Which of the following is the first line of output generated by the code segment above?
A. result: 0 C. result: 2
B. result: 1 D. None of the above.

7. Which of the following is the second line of output generated by the code segment above?
A. x: 7 C. x: 9
B. x: 8 D. None of the above.

8. Which of the following is the third line of output generated by the code segment above?
A. y: 11 C. y: 13
B. y: 12 D. None of the above.

9. Which of the following is the fourth line of output generated by the code segment above?
A. z: 5 C. z: 7
B. z: 6 D. None of the above.



Use the code segment below for problems 6 – 9

```
int x = 7;
int y = 10;
int z = 5;
int result = 0;

result = ++y - 10 || z - 5 && x++;
result += y++ - 11 || z++ - 5 && x++;
result += y + 1 > 11 && (z++ >= 6 || x++);

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

6. Which of the following is the first line of output generated by the code segment above?
- A. result: 0
 - B. result: 1
 - C. result: 2
 - D. None of the above.

```

int x = 7;
int y = 10;
int z = 5;
int result = 0;
    
```

```

result = ++y - 10 || (z - 5 && x++);
    
```

" " " - 10 = 1 = true

```

result += y++ - 11 || (z++ - 5 && x++);
    
```

" " " - 5

```

result += y + 1 > 11 && (z++ >= 6 || x++);
    
```

" " " - 5

$$y = \cancel{10}^{12}$$

$$\text{result} = \cancel{1}^2$$

" " " - 10 = 1 = true

```

result = ++y - 10 || (z - 5 && x++);
    
```

$$\text{res} = \text{res} + 0$$

Won't get evaluated

$$\underline{\underline{z++ + 2 - 5}}$$



```
result += y + 1 > 11 && (z++ >= 6 || x++);
```

expr1

expr2

GO CLASSES

$z = s$ $\begin{matrix} z++ \\ \text{---} \\ 5 \end{matrix}$ $+z - 5)$ $x++ +x$

accessing z again

before Seq. Point

\Rightarrow so you don't know
(compiler dep.)



```
int x = 7;
int y = 10;
int z = 5;
int result = 0;

result = ++y - 10 || z - 5 && x++;
result += y++ - 11 || z++ - 5 && x++;
result += y + 1 > 11 && (z++ >= 6 || x++);

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

7. Which of the following is the second line of output generated by the code segment above?

A. x: 7 C. x: 9
B. x: 8 D. None of the above.



```
int x = 7;
int y = 10;
int z = 5;
int result = 0;

result = ++y - 10 || z - 5 && x++;
result += y++ - 11 || z++ - 5 && x++;
result += y + 1 > 11 && (z++ >= 6 || x++);

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

8. Which of the following is the third line of output generated by the code segment above?
- A. y: 11
B. y: 12
C. y: 13
D. None of the above.



```
int x = 7;
int y = 10;
int z = 5;
int result = 0;

result = ++y - 10 || z - 5 && x++;
result += y++ - 11 || z++ - 5 && x++;
result += y + 1 > 11 && (z++ >= 6 || x++);

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

9. Which of the following is the fourth line of output generated by the code segment above?
- A. z: 5
 - B. z: 6
 - C. z: 7
 - D. None of the above.



C Programming

```
result = ++y - 10 || (z - 5 && x++);    && has higher precedence than ||
= 11 - 10 || (z - 5 && x++)
= 1 || (z - 5 && x++);
= True || don't care
= true
```

A blue arrow points from the text "y value will be 11 now" to the value "11" in the first line of code.

```
result += y++ - 11 || z++ - 5 && x++;
```

(see next page)



C Programming

```
result += y++ - 11 || z++ - 5 && x++;  
  
result += y++ - 11 || (z++ - 5 && x++);  
result += 11 - 11 || (z++ - 5 && x++); (y is 12 now)  
  
result += 0 || (z++ - 5 && x++);  
result += 0 || (5 - 5 && x++); (z is 6 now)  
  
result += 0 || (0 && x++);  
  
result += 0 || (0 && dont care);  
  
result += 0 || 0;  
  
result += false  
  
result += 0  
result = result + 0 = 1
```



C Programming

Here $y = 12$, $z = 6$, $result = 1$ and $x = 7$

```
result += y + 1 > 11 && (z++ >= 6 || x++);
result += 12+ 1 > 11 && (z++ >= 6 || x++);
result += 13 > 11 && (z++ >= 6 || x++);
result += true && (z++ >= 6 || x++);
result += true && (6 >= 6 || x++);           (z is 7 now)
result += true && (true || x++);
result += true && (true || dont care);
result += true

result = result +1
result = 1 +1 =2
```

Finally – $result = 2$, $x = 7$, $y = 12$ and $z = 7$



Question 10

```
int a=1, b=3, c=1;  
  
if( (a||c--) && (c&&b--) )  
    printf("%d\n",b);
```

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

$a = 1$

$b = 3$

$c = 1$



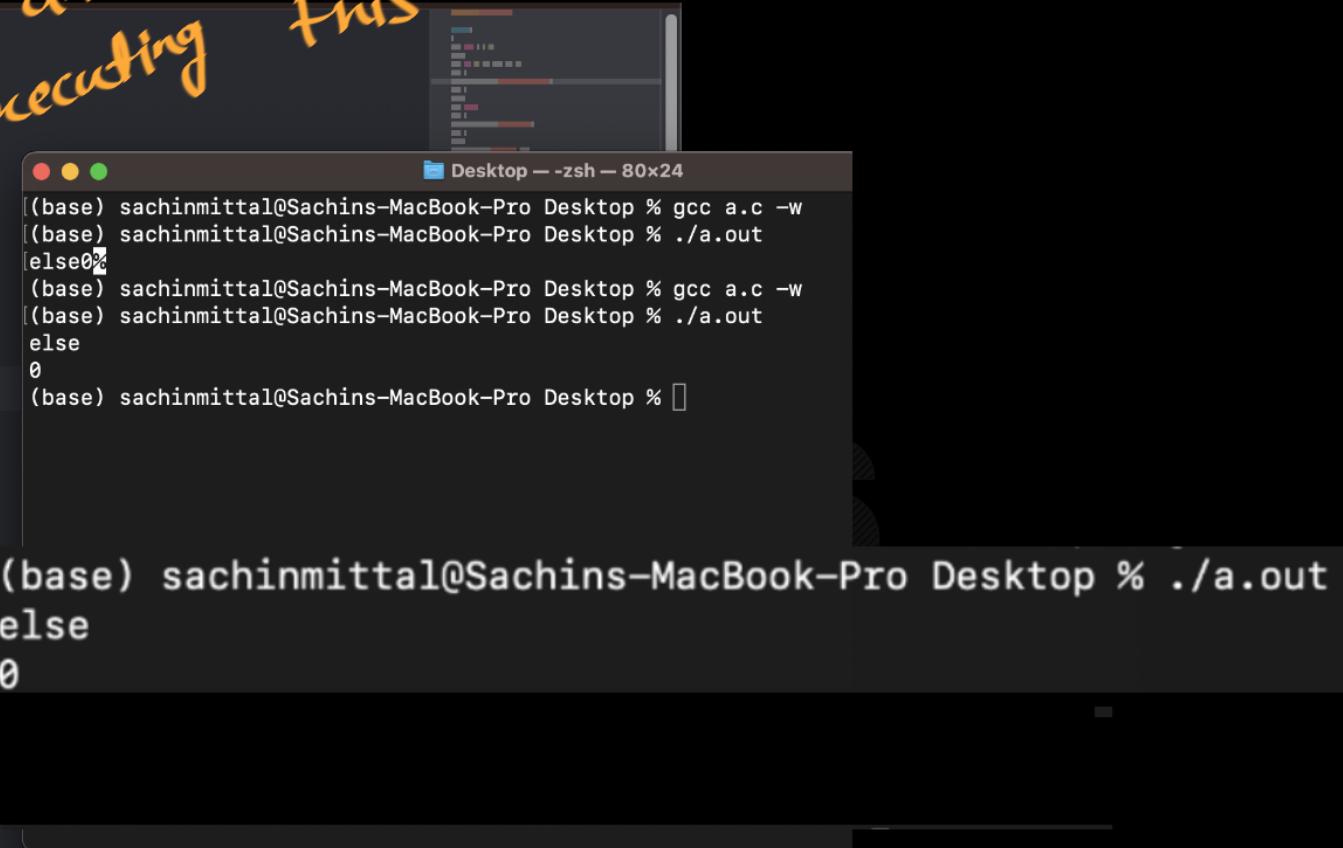


C Programming

```
main()
{
    int a = 0;
    if (0 && a++ == 0)
    {
        printf("Inside if");
    }
    else
    {
        printf("else");
    }
    printf("%d", a);
}
```

```
1 #include<stdio.h>
2
3 main()
4 {
5     int a = 0;
6     if (0 && (a++ == 0)
7     {
8         printf("Inside if\n");
9     }
10
11
12 else
13 {
14     printf("else\n");
15 }
16
17 printf("%d\n", a);
18 }
19 }
```

We are NOT
executing this



```
Desktop -- zsh - 80x24
(base) sachinmittal@Sachins-MacBook-Pro Desktop % gcc a.c -w
(base) sachinmittal@Sachins-MacBook-Pro Desktop % ./a.out
else0%
(base) sachinmittal@Sachins-MacBook-Pro Desktop % gcc a.c -w
(base) sachinmittal@Sachins-MacBook-Pro Desktop % ./a.out
else
0
(base) sachinmittal@Sachins-MacBook-Pro Desktop % 
(base) sachinmittal@Sachins-MacBook-Pro Desktop % ./a.out
else
0
```



Question 12

```
int i = 4, j = -1, k = 0, w, x, y, z ;
```

```
w = i || j || k ;  
x = i && j && k ;  
y = i || j && k ;  
z = i && j || k ;
```

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

- A. 1, 1, 1, 1
- B. 1, 1, 0, 1
- C. 1, 0, 0, 1
- D. 1, 0, 1, 1

$$w = \underbrace{(i \ ||\ j)}_{\text{True}} \ ||\ k ;$$

$$x = \underbrace{(i \ \&\& j)}_{\text{False}} \ \&\& k ;$$

$$y = \underbrace{i}_{\text{True}} \ || (j \ \&\& k) ;$$



Question 13

```
int main() {
    int i,j,k,l,m;
    //Line 1
    m = (i++&&j++&&k++) || l++;
    printf("i=%d j=%d k=%d l=%d m=%d\n",i,j,k,l,m);
}
```

Write down the output in each of the following cases when Line 1 is replaced by –

1. i = -1, j=-1, k=-1, l=2;
2. i = -1, j=-1, k=0, l=2;
3. i = 0, j=2, k=3, l=2; _____

1. i = -1, j=-1, k=-1, l=2;

won't get executed

```
//Line 1
m = (i++&&j++&&k++) || l++;
printf("i=%d    j=%d    k=%d    l=%d    m=%d\n", i, j, k, l, m);
}
```

True

i
j
k
l
m

0 0 1 3

2. i = -1, j=-1, k=0, l=2;

```
//Line 1
m = (i++&&j++&&k++) || l++;
printf("i=%d    j=%d    k=%d    l=%d
      m=%d\n", i, j, k, l, m);
}
```

0 0 3 1

Write down the output in each of the following cases when Line 1 is replaced by –

3. i = 0, j=2, k=3, l=2;

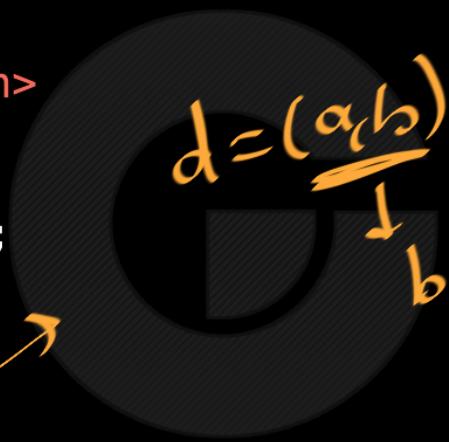
```
//Line 1
m = (i++&&j++&&k++) || l++;
printf("i=%d    j=%d    k=%d    l=%d    m=%d\n", i, j, k, l, m);
}
```

↓ ↓ ↓ ↓ ↓
1 2 3 3 3

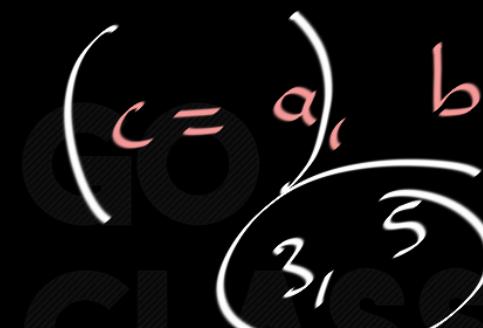
(i++ && j++) -&& k++) || l++;

Question 14

```
#include<stdio.h>
main()
{
    int a,b,c,d;
    a=3;
    b=5;
    c=a,b;
    d=(a,b);
    printf("c=%d d=%d", c, d);
}
```



↓ ↓
3 5



↓
3

$$\begin{aligned}d &= c = a, b \\ \Downarrow \quad c &= 3 \\ d &= 3\end{aligned}$$



Question 15

```
#include<stdio.h>
main()
{
    int a, b=2, c;
    a = 2*(b++);
    c = 2*(++b);
    printf("a = %d c = %d", a,c);
}
```

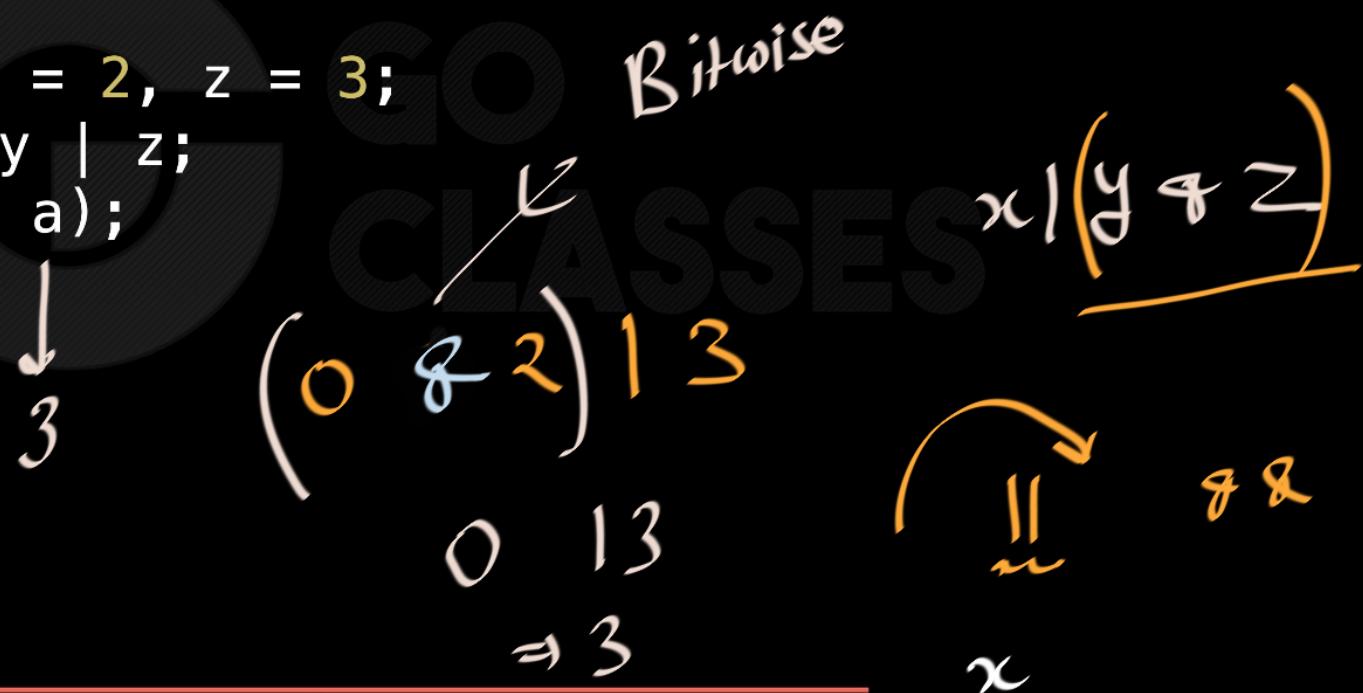
Annotations on the code:

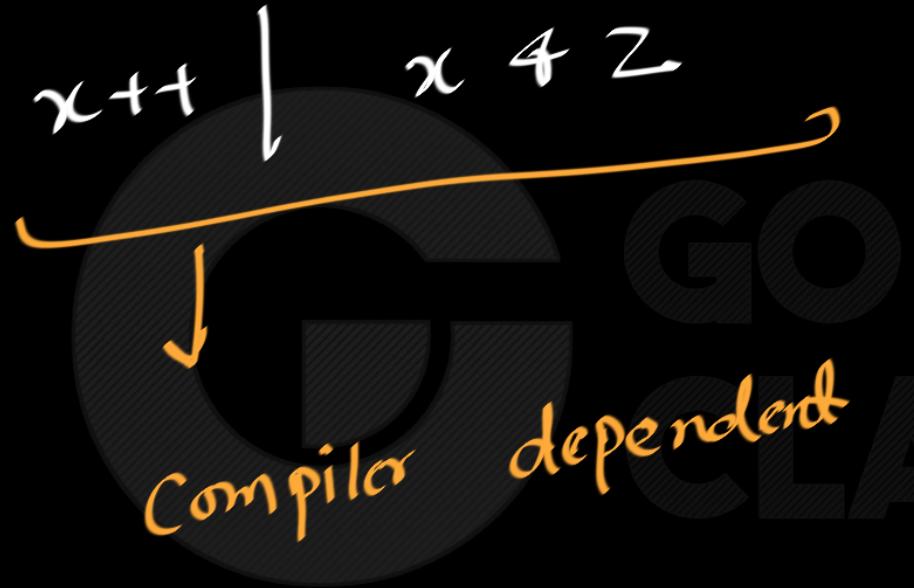
- A handwritten '3' is written above the variable 'c'.
- An arrow points from the handwritten '2' to the post-increment operator 'b++' in the assignment 'a = 2*(b++)'; another arrow points from this same '2' to the plus sign '+' in the expression.
- An arrow points from the handwritten '4' to the pre-increment operator '++b' in the assignment 'c = 2*(++b);'.
- A large handwritten '4' is written below the closing brace '}'.
- A handwritten '4448' is written below the printf statement.

Question 16

```
#include <stdio.h>
void main()
{
    int x = 0, y = 2, z = 3;
    int a = x & y | z;
    printf("%d", a);
}
```

- a) 3
- b) 0
- c) 2
- d) Run time error





$x++ || x \& z$

this is NOT
compiler depend.



Question 17

```
#include <stdio.h>
int main()
{
    int a = 10, b = 5, c = 3;
    b != !a;
    c = !!a;
    printf("%d\t%d", b, c);
}
```

- a) 5 1
- b) 0 3
- c) 5 3
- d) 1 1



Question 17

```
#include <stdio.h>
int main()
{
    int a = 10, b = 5, c = 3;
    b != !a;
}
```

- a) 5 1
- b) 0 3
- c) 5 3
- d) 1 1

b ≠ 3;
→ 5 ↗ 5

this is NOT assignment

Question 17

```
#include <stdio.h>
int main()
{
    int a = 10, b = 5, c = 3;
    c = !!a;
    printf("%d\t%d", b, c);
}
```

- a) 5 1
- b) 0 3
- c) 5 3
- d) 1 1

5 |
 ↓
 , , a
 , , a
 , , a
 , , a

$10 = \text{True}$





Question 18

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

bitwise XOR

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

$$\begin{array}{r} 1000010 \\ 000000011 \\ \hline 01 \end{array}$$



Question 19

STANFORD

For what is/are values program execute printf ?

```
#include <stdio.h>
int main()
{
    int i;
    scanf("%d", &i);
    if (!i == ~i) printf("same this time\n");
}
```

YOGA

for $i=0$ 

0 0 0 0 0 0 0 0 0 0

or
0 0 0 0 0 0 0 0 0 1

GO CLASSES



= = ~i

$\checkmark \quad i = -\sim^i$



$i = \text{middle}$

$0000 \dots 0 = \sim^i$

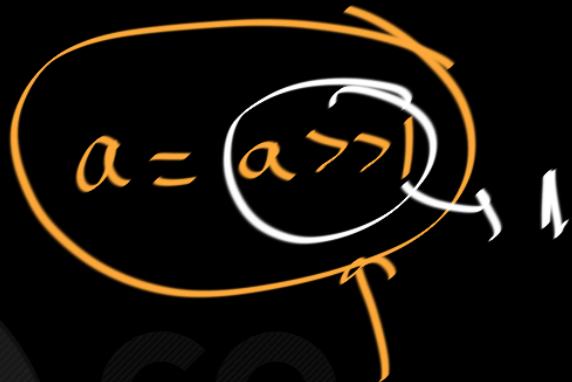
$00000 \dots 0 \} = \sim^i \quad i = 1111 \dots 10$

4-2



Question 20

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



we are NOT
writing this

Q

- a) 0
- b) 1
- c) 2
- d) No Output

a will not modify



Question 21

```
#include<stdio.h>
void main()
{
int k,i=50,j=100,l;

i=i|(j&&100);

k=i||(j||100);

l=i&(j&&100);

printf("%d %d\n",i,j);
printf("%d %d",k,l);
}
```

