



**AWESH ISLAM**  
**BUET, CSE**

# BATCH RECURSION

## C AND C++

PROGRAMMING MASTERCLASS

**FROM**  
**BASICS TO OBJECT-ORIENTED**

**Class-04**



**SHAROARE HOSAN EMON**  
**BME, BUET**



আমাদের সবগুলো ক্লাস দেখার জন্য ভিজিট করো  
<https://www.hsccrackers.com/>



SCAN ME

# Bitwise Operations

# Bitwise Not

একে প্রকাশ করা হয় ~ চিহ্ন দিয়ে

কাজ ও বৈশিষ্ট্য :

১) একটি unary অপারেটর

২) সব বিট কে ফ্লিপ করে দেয় যেমন ০০০০ কে ১১১১

# Bitwise Not

```
7
8  #include<stdio.h>
9
10 int main(){
11     int a,b;
12     a = 0;
13     b = ~a;
14     printf("a = %d, b = %d\n",a,b);
15     a = 1;
16     b = ~a;
17     printf("a = %d, b = %d\n",a,b);
18 }
19
```

```
a = 0, b = -1
a = 1, b = -2
Program ended with exit code: 0
```

# 2's Complement

- ১) Computer এ ঋণাত্মক সংখ্যা প্রকাশ করা হয় এর মাধ্যমে।
- ২) প্রথম ১ পাওয়া পর্যন্ত বিটগুলো ঠিক রেখে এর পরের বিটগুলো ফ্লিপ করে দিতে হয়।

# 2's Complement Example

1byte Binary: 01000010 = 66

2's complement : 10111110 = -66

Explanation of output:

In computer memory 0 = 00000000

When we perform  $\sim 0 = 11111111$

2's complement of 11111111 is 00000001 which is 1 so 11111111 is -1

similarly 1 in computer memory is 00000001

$\sim 1$  is 11111110

2's complement is 00000010 which is binary of 2

So 11111110 is -2

# Shift Operator

Left Shift :  $\ll$  n

Right Shift :  $\gg$  n

কাজ : n ঘর ডানে বামে বিট শিফট করা।



# Left Shift Operator Example

1 = 00000001

1<<1 = 00000010 = 2

1<<2 = 00000100 = 4

10 = 00001010

10<<1 = 00010100 = 20

10<<3 = 01010000 = 80

এক বার লেফট শিফট করা মানে ২ দিয়ে গুণ করা আর n বার লেফট শিফট করা মানে  $2^n$  দিয়ে গুণ করা

# Left Shift Operator Example

```
8  #include<stdio.h>
9
10 int main(){
11     int a,b,c;
12     a = 10;
13     b = 10<<1;
14     c = 10<<3;
15     printf("%d %d %d\n",a,b,c);
16 }
```



**10 20 80**

**Program ended with exit code: 0**



# Left Shift Operator Example

```
8  #include<stdio.h>
9
10 int main(){
11     int a;
12     printf("Please enter a number: ");
13     scanf("%d",&a);
14     int n;
15     printf("How many bits you want to left shift it: ");
16     scanf("%d",&n);
17     printf("%d\n",a<<n);
18 }
```

# Right Shift Operator Example

1 = 00000001

1>>1 = 00000000 = 0

10 = 00001010

10<<1 = 00000101 = 5

10<<3 = 00000001 = 1

8 = 00001000

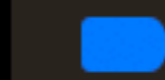
8<<3 = 00000001 = 1

এক বার রাইট শিফট করা মানে ২ দিয়ে ভাগ করা আর n বার রাইট শিফট করা মানে  $2^n$  দিয়ে ভাগ করা



# Right Shift Operator Example

```
8  #include<stdio.h>
9
10 int main(){
11     int a,b,c;
12     a = 10;
13     b = 10>>1;
14     c = 10>>3;
15     printf("%d %d %d\n",a,b,c);
16 }
17
```



**10 5 1**

**Program ended with exit code: 0**

# Right Shift Operator Example

```
7
8  #include<stdio.h>
9
10 int main(){
11     int a;
12     printf("Please enter a number: ");
13     scanf("%d",&a);
14     int n;
15     printf("How many bits you want to left shift it: ");
16     scanf("%d",&n);
17     printf("%d\n",a>>n);
18 }
19 |
```



# Right Shift Operator Example

```
7
8  #include<stdio.h>
9
10 int main(){
11     int a;
12     printf("Please enter a number: ");
13     scanf("%d",&a);
14     int n;
15     printf("How many bits you want to left shift it: ");
16     scanf("%d",&n);
17     printf("%d\n",a>>n);
18 }
19 |
```

# And Operator Truth Table

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1



# And Operator Example

7 = 00000111

9 = 00001001

---

7 & 9 = 00000001 = 1

# And Operator Example

```
8  #include<stdio.h>
9
10 int main(){
11     int a,b;
12     printf("Please enter two numbers you want to and : ");
13     scanf("%d %d",&a,&b);
14     printf("%d & %d = %d\n",a,b,a&b);
15 }
```

Please enter two numbers you want to and : 7 9  
7 & 9 = 1  
Program ended with exit code: 0

# Or Operator Truth Table

A	B	A   B
0	0	0
0	1	1
1	0	1
1	1	1



# Or Operator Example

7 = 00000111

9 = 00001001

---

7 | 9 = 00001111 = 15

# Or Operator Example

```
8  #include<stdio.h>
9
10 int main(){
11     int a,b;
12     printf("Please enter two numbers you want to or : ");
13     scanf("%d %d",&a,&b);
14     printf("%d | %d = %d\n",a,b,a|b);
15 }
```

```
▶
Please enter two numbers you want to or : 7 9
7 | 9 = 15
Program ended with exit code: 0
```

# XOR Operator Truth Table

A	B	$A \wedge B$
0	0	0
0	1	1
1	0	1
1	1	0



# XOR Operator Example

7 = 00000111

9 = 00001001

---

$7 \wedge 9 = 00001110 = 14$

# XOR Operator Example

```
8  #include<stdio.h>
9
10 int main(){
11     int a,b;
12     printf("Please enter two numbers you want to xor : ");
13     scanf("%d %d",&a,&b);
14     printf("%d ^ %d = %d\n",a,b,a^b);
15 }
```

Please enter two numbers you want to xor : 7 9  
7 ^ 9 = 14  
Program ended with exit code: 0

# Odd & Even With &

10	0000 1010
11	0000 1011
12	0000 1100
13	0000 1101
14	0000 1110
15	0000 1111
16	0001 0000
17	0001 0001
18	0001 0010
19	0001 0011
20	0001 0100
21	0001 0101
22	0001 0110



# Odd & Even With &

```
8  #include<stdio.h>
9
10 int main(){
11     int n;
12     printf("Please Enter a number: ");
13     scanf("%d",&n);
14     if(n & 1){
15         printf("%d is odd\n",n);
16     }
17     else{
18         printf("%d is even\n",n);
19     }
20 }
```

# TO LOWERCASE

32 = 00100000

Capital A	A	Shift A	65	01000001	41
Capital B	B	Shift B	66	01000010	42
Capital C	C	Shift C	67	01000011	43
Capital D	D	Shift D	68	01000100	44
Capital E	E	Shift E	69	01000101	45
Capital F	F	Shift F	70	01000110	46
Capital G	G	Shift G	71	01000111	47
Capital H	H	Shift H	72	01001000	48
Capital I	I	Shift I	73	01001001	49
Capital J	J	Shift J	74	01001010	4A

# To Lowercase

```
8  #include<stdio.h>
9
10 int main(){
11     char c;
12     printf("Please Enter a uppercase alphabet: ");
13     scanf("%c",&c);
14     printf("Converting it to lowercase: %c\n",c|32);
15 }
16
```

# TO LOWERCASE

95 = 0101 1111

Lower-case A	a	A	97	01100001	61
Lower-case B	b	B	98	01100010	62
Lower-case C	c	C	99	01100011	63
Lower-case D	d	D	100	01100100	64
Lower-case E	e	E	101	01100101	65
Lower-case F	f	F	102	01100110	66
Lower-case G	g	G	103	01100111	67
Lower-case H	h	H	104	01101000	68
Lower-case I	I	I	105	01101001	69
Lower-case J	j	J	106	01101010	6A



# To Uppercase

```
8  #include<stdio.h>
9
10 int main(){
11     char c;
12     printf("Please Enter a lowercase alphabet: ");
13     scanf("%c",&c);
14     printf("Converting it to uppercase: %c\n",c & 95);
15 }
```

# Check if Power of 2

**8 = 00001000**

**7 = 00000111**

**16 = 00010000**

**15 = 00001111**

**32 = 00100000**

**31 = 00011111**

# Check if Power of 2

```
8  #include<stdio.h>
9
10 int main(){
11     int n;
12     printf("Please Enter a number: ");
13     scanf("%d",&n);
14     if(n && !(n & (n-1))) printf("%d is a power of 2\n",n);
15     else printf("%d is not a power of 2\n",n);
16 }
```

# Count Number of ones

```
8  #include<stdio.h>
9
10 int main(){
11     int n, count = 0;
12     printf("Enter a number: ");
13     scanf("%d",&n);
14     count = 0;
15     while (n) {
16         if(n & 1) count++;
17         n = n >> 1;
18     }
19     printf("Number of 1 bits in %d is : %d\n",n,count);
20     return 0;
21 }
```

# Check if 2 numbers equal

$$A \wedge A = 0$$

```
8  #include<stdio.h>
9
10 int main(){
11     int n, m;
12     printf("Enter two numbers: ");
13     scanf("%d %d",&n,&m);
14     if(n ^ m) printf("The numbers are not equal\n");
15     else printf("The numbers are equal\n");
16 }
```



# Find Unique

$$A \oplus 0 = A$$

```
8  #include<stdio.h>
9
10 int main(){
11     int arr[7] = {1,2,3,4,3,2,1};
12     int unique = 0;
13     for(int i = 0;i < 7;i++){
14         unique ^= arr[i];
15     }
16     printf("%d is the unique number in array\n",unique);
17 }
```

```
4 is the unique number in array
Program ended with exit code: 0
```