| 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 2147483648 | 1073741824 | 536870912 |  |  |  |  |  |  |  |  |  |  |  |  | 65538 | 32768 | 16384 | 8192 | 4096 | 2048 | 1024 | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 3 | 2 | 1 |

Tricks

Is suppose all rights of any number is on then its sum is 1 less then the number.

Like

32 16 8 4 2 1

0 1 1 1 1 1 -> 31

128 64 32 16 8 4 2 1

0 1 1 1 1 1 1 1 -> 127

128 64 32 16 8 4 2 1

0 1 1 1 1 1 0 1 -> 127-2 - 125

Now we know value of 2^31 = 2147483648

If all right is on

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -> 2147483647 (this is the max possibility)

Find binary of 66

So 64 will be on and 1 is left we will make 2 on so it will be 66. -> 1000010

Example of overload:

We have printed value over the range

X = 2147483659;

printf(“%d”, x)

Now as per above trick we wil find nearest number

Which is 2147483648 (2^31)

Means 32nd bit we can see here

| 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Now we will find difference 2147483659 - 2147483648 = 11;

Now we will place in table

As per trick we know

if the nearest number we got then that number will be on and other will be off.

We are not taking sign bit as of now.

So 10000000000000000000000000000000 this is of 2147483648 now there is 11 difference

So

2147483648 1073741824 536870912 …. 32 16 8 4 2 1 (8+2+1 = 11)

1 0 0 …….. 0 0 1 0 1 1 = 2147483659

10000000000000000000000000001011 is Binary of 2147483659

Now we will give this value to printf

Printf will consider 1st value as sign bit to it will make value -ve and find its 2’s compliment

2’s -> 01111111111111111111111111110100

Now add 1 in 2s

01111111111111111111111111110100

1

—--------------------------------------------

01111111111111111111111111110101 Find decimal of this number

As we know if all is on means value is 2147483647 now 2nd and 4th is off means

2147483647 - 2^3 - 2^1

2147483647 - 8 - 2

2147483647 - 10 =

2147483637 with a Negative sign.

#include<stdio.h>

int main()

{

int x;

x = 2147483659;

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

}

/\*

Output:

-2147483637

\*/

Hence prove we can find the values, the values we provide out of range are not garbage values.

Take another Example

X = -2147483660

Find nearest 2147483648 and its binary is

| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Now find the difference between values

2147483660 - 2147483648 = 12

2147483648 1073741824 536870912 …. 32 16 8 4 2 1 (8+4 = 12)

1 0 0 …….. 0 0 1 1 0 0 = 2147483660

| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Now we its negative value we find its 2s compliment

And we have to add 1 in that

| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

This is the value will be given of x which will be given to printf

Now printf will calculate

1st bit 0 means positive

All bits are on except 0th, 1st,3rd

2147483647 - 2^3 - 2^1 - 2^0

2147483647 - 8 - 2 - 1

214748364 - 11

2147483636 (Value will be positive because 1st bit was 0)

#include<stdio.h>

int main()

{

int x;

x = -2147493660;

printf("%d", x);

return 0;

}

/\*

output:

2147473636

\*/

One more case

If while converting from decimal to binary it limit exceed with 32 then printf will only consider right 32 bits.

Example

#include<stdio.h>

int main(){

int x;

x = 394147483648;

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

return 0;

}

Binary of 394147483648 is of length 40

0101101111000101000001010101000000000000

**Now we will consider only 32 bits**

~~01011011~~11000101000001010101000000000000

11000101000001010101000000000000

1st value is 1 means value will be Negative

Now find 2s

11000101000001010101000000000000

00111010111110101010111111111111

* 1

—-----------------------------------------------

00111010111110101011000000000000

This is a binary of 989507584

/\*

output:

-989507584

\*/

**Unsigned INT**

#include<stdio.h>

int main(){

unsigned int x; // 1st is not sign bit

x = 2147483659;

printf("%d\n",x); // %d means 1st sign and rest data

printf("%u\n", x); // %u stands for unsigned int.

return 0;

}

/\*

output

-2147483637

2147483659

\*/

As we know value will not going to be negative in that case range will be increase.

When its signed then range is

-2147483648 to 2147483647 means -2^31 to 2^31-1

But in case of unsigned int

0 to 2^32-1