

3

2018 December

Monday

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Algorithm - Graphical Representation  
of your Algo.

Pseudocode - Generic way of Representing  
your Algo in textual form  
like - English

Using Namespace std;  
Means -

4

There are multiple namespaces, in each  
namespace cout has different  
Tuesday different implementation.

But we gonna use std (standard)  
namespace

namespace namespace namespace

cout has  
diff  
impl  
mutat  
ion

cout  
has  
differ  
impl.

''

std

for signed -  $-2^{31}$  to  $2^{31}-1$

for Unsigned - 0 to  $2^{32}$

January 2019						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

December 2018

Wednesday

5

Q bool data type take 1 byte space in memory.

But as we know bool represent only  $\downarrow$  1 & 0  
true      false

so why bool not take 1 bit.

Ans

because 1 byte is minimum memory space.

And 1 bit is not sufficient find addresses, as we know every address stored in 8 bits.

smallest Addressable memory unit is 1 byte.

6

no. of bits    1  $\rightarrow$  1  
 $2^1 - 1$              $\leftarrow$  2  $\rightarrow$  10  
                        3  $\rightarrow$  11       $\leftrightarrow$  in 2 bit (max = 3)  
                        4  $\rightarrow$  100  
                        5  $\rightarrow$  101

no. of bits    6  $\rightarrow$  110  
 $2^6 - 1$              $\leftarrow$  7  $\rightarrow$  111       $\rightarrow$  in 3 bit (max = 7)  
                        8  $\rightarrow$  1000  
                        9  $\rightarrow$  1001

no. of bits    15  $\rightarrow$  1111       $\rightarrow$  in 4 bit (max = 15)  
 $2^{15} - 1$              $\leftarrow$  16  $\rightarrow$  11111       $\rightarrow$  in 5 bit (max = 31)

January 2019						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

December 2018

Saturday

1

\* Four short datatype  $\rightarrow$  2 byte  $\rightarrow$  16 bits

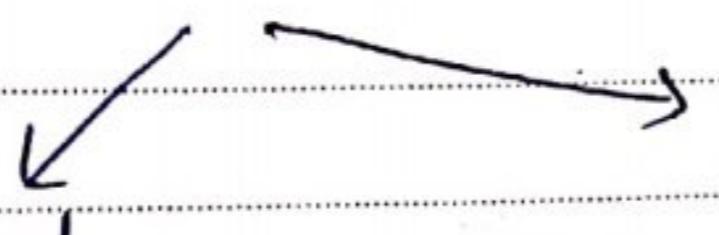
Total Combination =  $2^{16}$

four unsigned  $\rightarrow$  0 to  $2^{16}-1$

four signed  $\rightarrow$   $-2^{15}$  to  $2^{15}-1$

\* Four character datatype  $\rightarrow$  1 byte  $\rightarrow$  8 bits

Total combination =  $2^8$



signed  
 $-2^7$  to  $2^7-1$       Unsigned  
 $0$  to  $2^8-1$

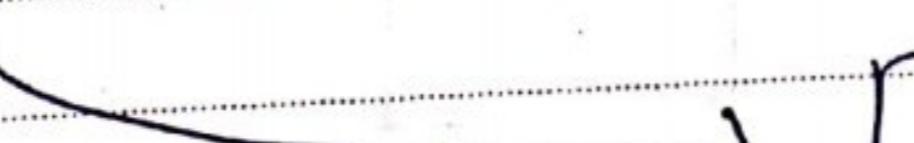
Sunday

2

Implicit Type Conversion (by Automatic)

char ch = 'A';  
cout << ch;  $\rightarrow$  a

Explicit Type Conversion (by itself)

Syntax float e = (float) 2;  
  
(type) expression

29

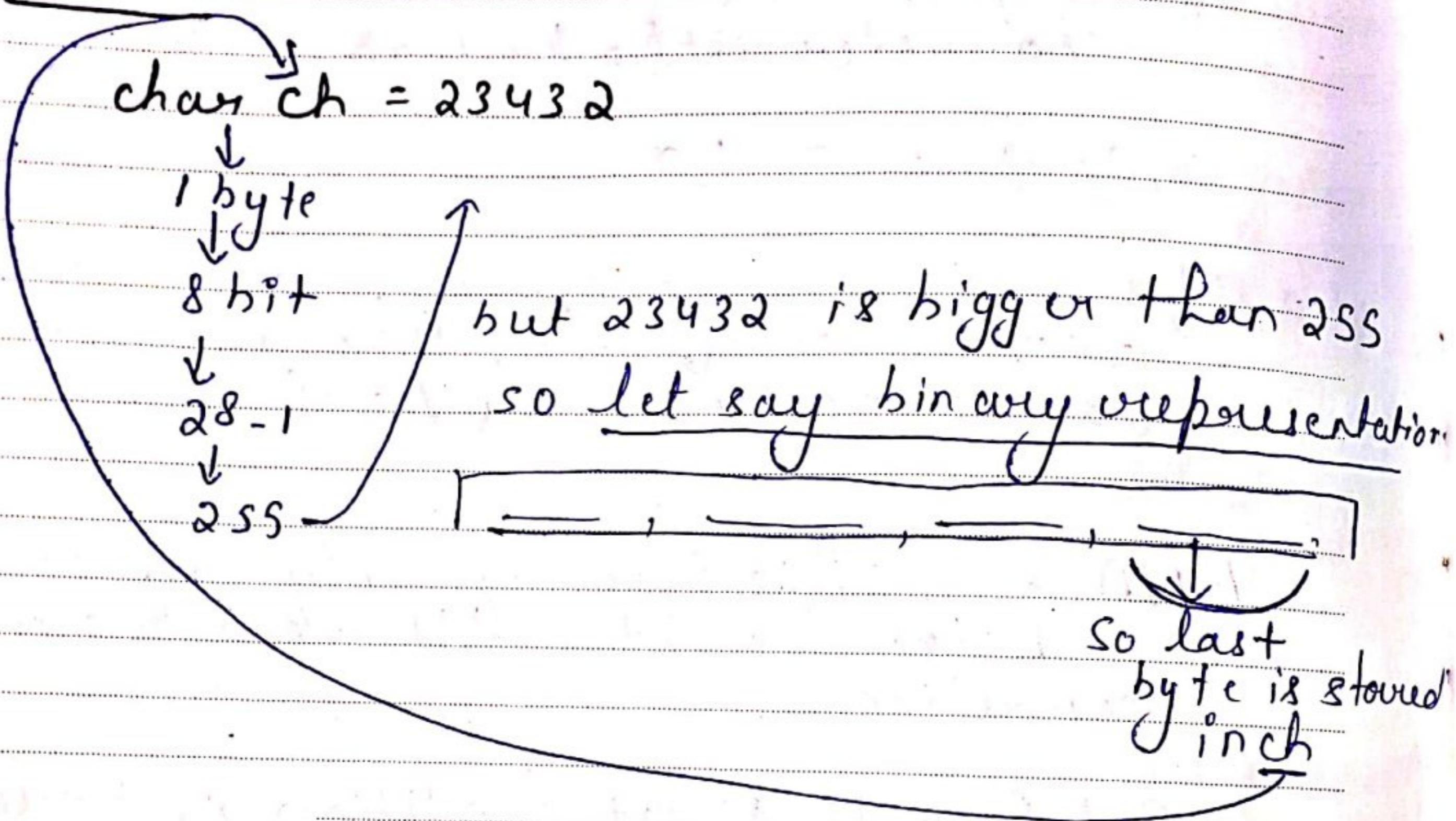
2018 November

Thursday

October

2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Doubt

30

Friday

Important

In tutorial, During value detection if we put space, enter, tab, so cin ignores.

Means space, enter, tab press ignore next value input line by line.

This is called white space character.

But if we want to read spaces, enter, tab.

So we use `cin.get()`

Ex `int a, b`  
`a = cin.get();`

December 2018						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

November 2018

Tuesday

27

space

If we give  $1 \backslash 2$ , so it's ne '1' as a character whakidga

so, output = 49

If we give only space, so this space act as character, this integer value achieved by Ascii table.

AND  $\rightarrow$  In this case, if only one condition is false, so it won't check another condition

But for, if first condition is true so, it also check second condition also.

Wednesday 28

OR  $\rightarrow$  In this case, only one of the conditions must be true, so if won't check  $a > 2$

25

2018 November

Sunday

October

2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Bitwise operators → It performs own operation on Bit level Q81

Left shift me Agar hum -ve value ke bit ka left shift krenge to uska MSB 1 to 0 Hone se, wo -ve to +ve bn jaega.

But in case of Right shift -ve value simply divide by 2

int a = -12

a = a >> 1

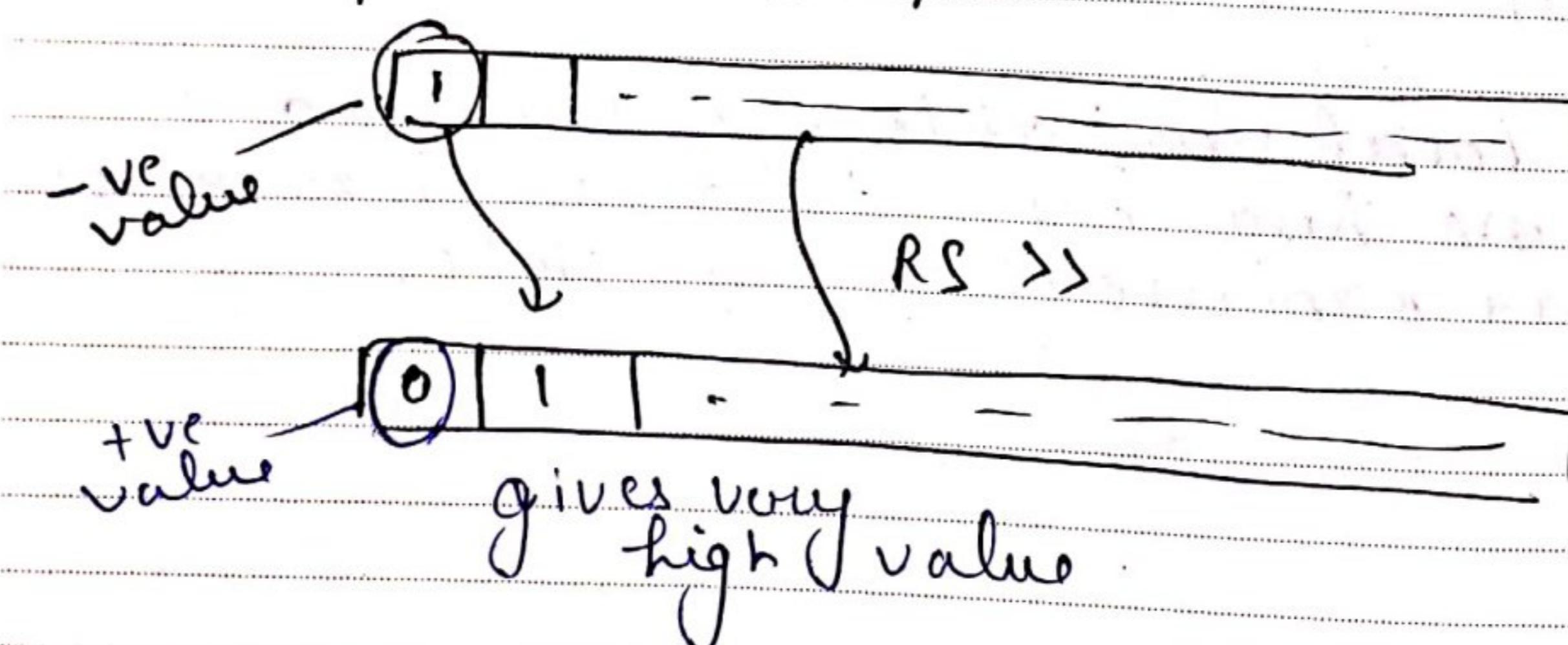
cout << "the value of a is :" << a << endl;

26

Monday

Output -6

Important Concept



Note - Jitna MSB ke pse hota hai utna wo bdi value hota hai, aur jitna wo lsh ke pse hota hai utna choti value hota hai

December 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

November 2018

Friday

23

## In variable scoping -

- \* Redefinition of any variable is not allowed.
- \* But variable ki value updation is allowed.

int b = 5;  
b = 10; ✓

int b = 15 X  
cout << b << endl → 10 Value.

Note - Global variable int main()  
se phle define kerte hai, Saturday 24  
aur usko hum kahi bhi use kar sakte  
hai, int main aur in any function.

But Local variable jis function <sup>a</sup> it  
define huwai, usko us-block ke  
baray kahi use nhi kar sakte.

21

2018 November

Wednesday

October

2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

We can use both `int main()`, as well as `void main()`

`void main()`

{

\_\_\_\_\_

}

\_\_\_\_\_

`int main()`

{

\_\_\_\_\_

}

`return 0;` → this is compulsorily if we use `int main()`.

## Function Call stack

22

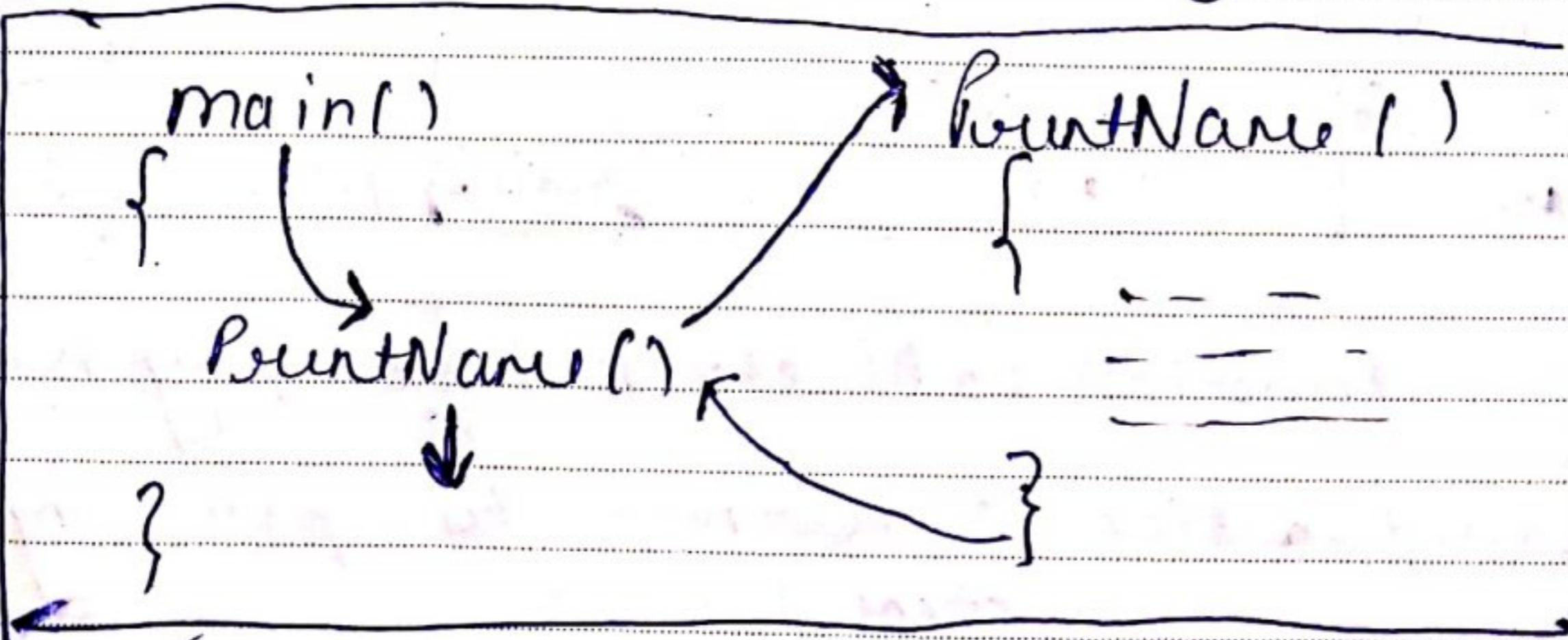
Thursday

→ indicates

- Function call
- kis function recursive function  
ko call kia hai
- function k hon-kon se  
local variable hai
- function kya return krega

stack  
LIFO

PointName  
main



December 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31				1	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

November 2018

Monday

19

Function fully execute lone ke hdd wo function call stack se automatically pop ho jate hai.

Ex-

```
int main() { void printa(int a) { void printb(int b) { int a=5; int b=3; cout<<a; cout<<b; return 0; } printb(b); }
```

successfully executed

Tuesday

20

After successfully executed function auto-matically, pop from stack.

After return 0

function call stack is empty Now

Main function return 0 to operating system.

17

2018 November

Saturday

October

2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Function call our function invoke is  
Same thing.

# Pass by Value →

14 as a  
copy array  
Lai.

main()

int num = 12

12  
num

num++

13  
num

14  
num

18

++num  
Sunday

Point(num)

--num 13  
num

cout << num → 13

}

void print(int num)

{

14  
num

++num;

15  
num

num++;

16  
num

cout << num; → 16

--num

15  
num

In Pass by value, from original value at  
Address na dekhi wo value as a  
copy form re de te Lai, jisse  
wo jo hi operation copy ke

December 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31				1	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29						

November 2018

Thursday

15

per own karte hai, wo hame original  
42 milta.

Both act as different memory location.

Note -> function declare & function define are different things,

Jaha bhi hum function ko call karne  
chahiye, at least uske swa

us function ka declaration uske  
calling se pehle hona chahiye.

Bhalo ham chaye to, usko Friday 16  
define int main ke bold bhi kar  
sakte hai, but declaration invoke  
se behle.

13

2018 November

Tuesday

October

2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

## What is Number System

Method to represent numeric values or quantities using different digit.

### Decimal Number System -

The decimal no. system has base 10.

It uses digit from 0 to 9.

Base - it is the number of symbols (digit) a number system uses.

14

Wednesday

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

} 10 no. of digit to represent Decimal (no.) system.

## Binary Number System

- Binary Number System uses base 2
- It uses only two digit i.e 0 and 1

December 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31				1	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

November 2018

Sunday

11

## Computer storage

CPU → CPU stores value in form of Binary  
Memory → Memory stores address  
in form of Binary

Power →

off    on    on    off    on    off

0    1    1    0    1    0  
↑    ↑    ↑    ↑    ↑    ↑  
GBits CPU

Ones

Jitne no. of bit  
hai utne no. of bits  
CPU has hai

Monday

12