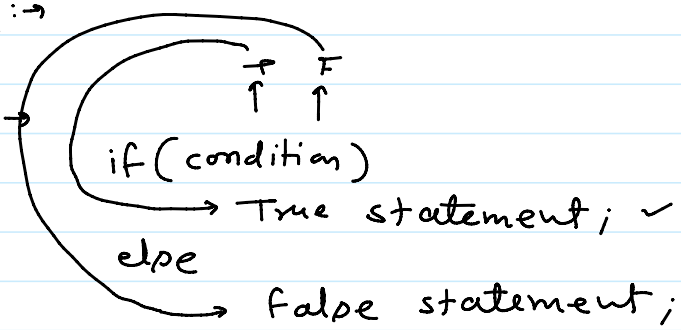
DAY 02

05 April 2024 01:07 PM

Condition :→

if else :→



Max of 2 Nos :→

```
a, b

if (a>b)
    printf ("%d", a);
else
    printf ("%d", b);
```

```
if ( condition )
{
    True statement1;
    True statement2;
}
else
{
    false statement1;
    false statement2;
}
```

find n^{th} bit of a no -

a | 175 | n | 3 | nth

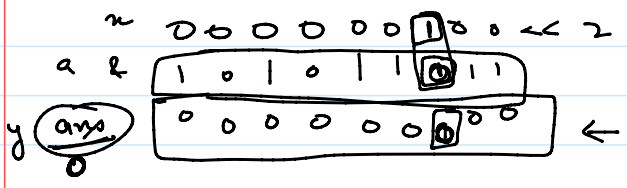
(175) → 1 0 1 0 1 1 1 1

x | 0 0 0 0 0 1 0 0

$x = 1 << (n-1)$

x | 0 0 0 0 0 1 0 0

x = 1 << (n-1)




```
int a, n, x, y;
input → a [175]
        n [3]
x = 1<<(n-1);
y = a & x;
if (y == 0)
    print("0");
else
    print("1");
```

zero (F)
non zero (T)

```
if (y)
    print("1");
else
    print("0");
```

Reverse nth bit of no -

a [171] [175]

n [3]

nth

a → 1 0 1 0 1 1 1 1

^ x → 0 0 0 0 1 0 0

a 1 0 1 0 1 0 1 1

☐ ^ 0 ⇒

☐ ^ 1 ⇒

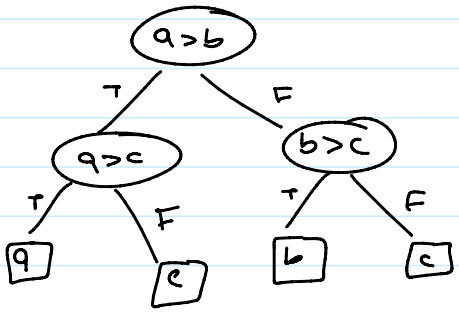
```
x = 1<<(n-1);
a = x ^ a;
```

Nested if else :-

```
if (cond1)
{
    if (cond2)
        True;
    else
        false;
}
else
```

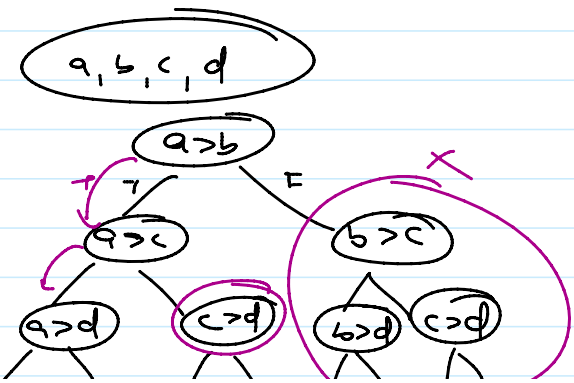
```
L3          false;
else
{       if (cond3)
            True;
        else
            false;
}
```

max of 3 Nos :→ a, b, c



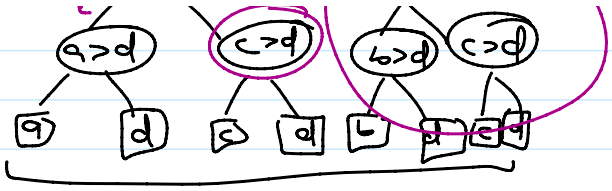

```
int a, b, c, max;
prin ———— %;
sca ———— (&a, &b, &c);

if(a>b)
{       if(a>c)
            max = a;
        else
            max = c;
}
else
{       if (b>c)
            max = b;
        else
            max = c;
}

printf ("%d", max);
```

		cond	stmt
max of 2	→	1	2
max of 3	→	3	4
max of 4	→	7	8
max of 5	→	15	16



max of 4 → 7

max of 5 → 15 16



③ only if :-

syntax → if (condition)
True stmt ;

max of 4 Nos :→ only if

a, b, c, d
15 2 7 3

max #7

```
max = a;
if (b > max)
    max = b;
```

max of 2 Nos

→ `if (c > max)`
`max = c;`

max of 3 Nos

```
if (d > max)
    max = d;
```

max of 4 Nos

→ `printf ("%d", max);`

ASCII code :→

Binary
0, 1

int → 0, 1
float → 0, 1
char 'a' → code (int) → 0, 1

25 / 2 = 12 rem 1
12 / 2 = 6 rem 0
6 / 2 = 3 rem 0
3 / 2 = 1 rem 1
1 / 2 = 0 rem 1

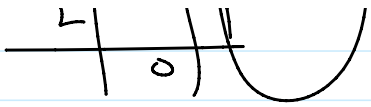
1 1 0 0 1

ASCII

25 − 16 = 9
9 − 8 = 1
1 − 1 = 0

2^5 2^4 2^3 2^2 2^1 2^0
32 16 8 4 2 1
0 1 1 0 0 1

ASCII − 7 → 128 Char



ASCII - 7 → 128 char

ASCII - 8 → 256 char

unicode → 65535 char wchar_t → 2 bytes

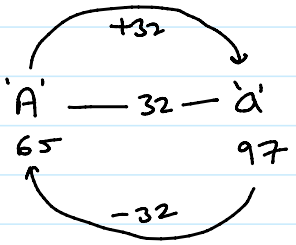
char (%c)			code (%d)
A	- Z	→	65 - 90
a	- z	→	97 - 122
'0'	- '9'	→	48 - 57
space		→	32

printf ("%d", 'A'); 65

code char

cout << 'A'; → 65

cout << int ('A');



Convert a char from upper to lower case -

66+32 → 98

ch [B b]

65-90 → A → a

r → x

5 → x

@ → x

```
if (ch >= 65 && ch <= 90)
    ch = ch + 32;
printf ("%c", ch);
```

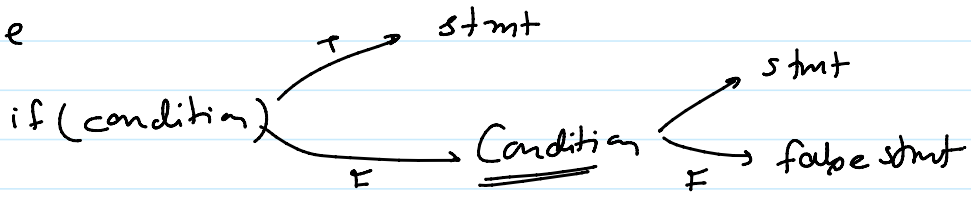
A → a

r → R

ch [A a]

```
if (ch >= 65 && ch <= 90)
    ch = ch + 32;
else if (ch >= 97 && ch <= 122)
    ch = ch - 32;
printf ("%c", ch);
```

elpe if elpe



W.A.P to print Grade of a student if percentage of the student is given —

① per >= 90 ⟶ A
② per >= 70 & < 90 ⟶ B
③ per >= 50 & < 70 ⟶ C
④ per < 50 ⟶ D

```
float p;
prin ———— );
scq  ———— &p);
if (p>=90)
        printf ("A");
else if (p>=70)
        printf("B");
else if (p>=50)
        printf("C");
else
        printf("D");
```

print name of a day no of the rp given —


```
int day;
printf( ———— );
scanf ( ———— &day);
if (day==1)
        printf ("mon");
else if (day==2)
        printf ("Tues");
else if (day==3)
```

1 — mon
2 — Tue
3 — wed
⋮
7 — Sun
other ⟶ Invalid

```
elseif(day == 7)
        printf ("Sunday");
else
        printf ("Invalid");
```

switch case :-

Syntax :-

```
switch (number)
{
    case value:
        statement;
        break;
    case value:
        stmt;
        break;
    default:
        stmt;
}
```

```
switch ( — )
{
}
```

```
switch ( — )
{
    case 1: =
        switch ( — )
        {
        }
}
```

Rules of switch case:-

1.The expression provided in the switch should result in a constant integer value otherwise it would not be valid.

note :- Case value must be constant.

e.g. Constant expressions allowed

switch(1+2+23)

switch(1*2+3%4)

2.Duplicate case values are not allowed.

3.The default statement is optional.

4.The break statement is used inside the switch to terminate a statement sequence.

5.The break statement is optional.

6. case sequence can be anything i.e one can create default in beginning etc.

7.Nesting of switch statements are allowed, which means you can have switch statements inside another switch. However nested switch statements should be avoided as it makes program more complex and less readable.

month no ⟶ Print no. of days in the month

month no ⟶ Print no of days in the month

1 — 31 day
2 — 28/29 days
3 — 31 —
4 — 30
⋮
12 — 31
other — invalid

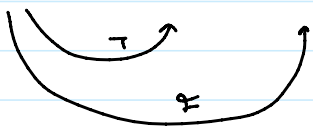
```
//max of 4 nos
#include <stdio.h>
int main()
{
	int day;
	printf("Enter no of a day:");
	scanf("%d",&day);
	switch(day)
	{
		case 1:
			printf("Monday");
			break;
		case 2:
			printf("Tuesday");
			break;
		case 3:
			printf("Wednesday");
			break;
		case 4:
			printf("Thursday");
			break;
		case 5:
			printf("Friday");
			break;
		case 6:
			printf("Saturday");
			break;
		case 7:
			printf("Sunday");
			break;
	}
	return 0;
}
```

```
#include <stdio.h>
int main()
{
	int m;
	printf("Enter no of a month:");
	scanf("%d",&m);
	switch(m)
	{
		case 1:
		case 3:
		case 5:
```

```
                case 7:
                case 8:
                case 10:
                case 12:
                        printf("31 Days");
                        break;
                case 2:
                        printf("28|29 Days");
                        break;
                case 4:
                case 6:
                case 9:
                case 11:
                        printf("30 Days");
                        break;
                default:
                        printf("Invalid");
        }
        return 0;
}
```

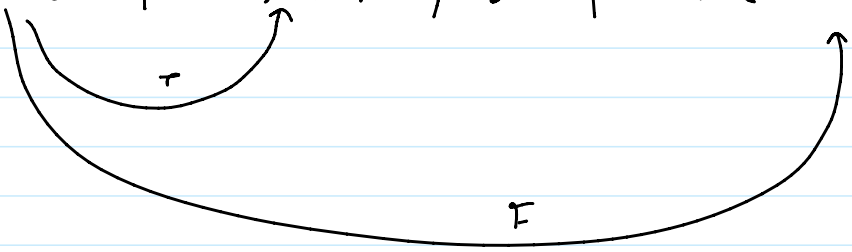
Conditional operator (?:) → (Ternary op)

Syntax → cond ? True : falp ;



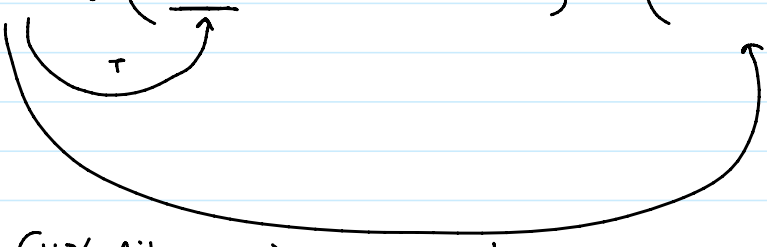
max of 2 Nos —

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



Nested :- max of 3 → a, b, c

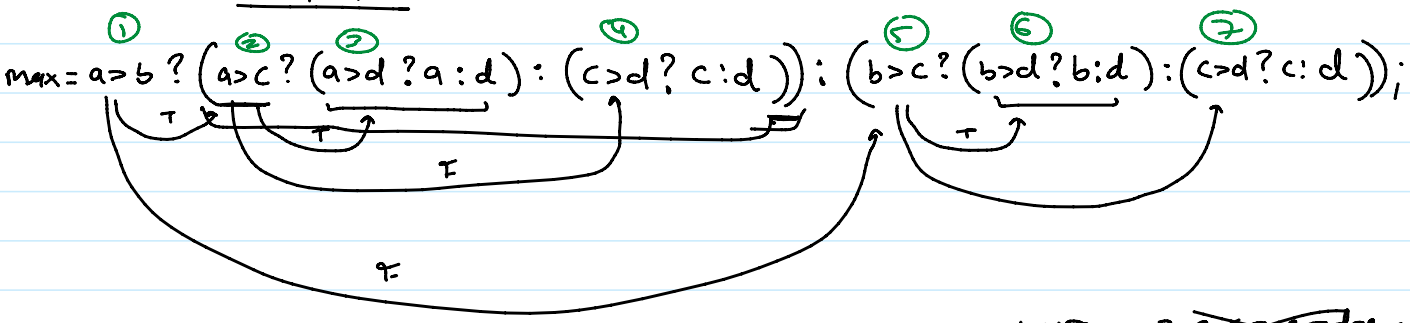
max = a>b ? (a>c ? a : c) : (b>c ? b : c);



printf("%d", max);

max of 4 nos =>

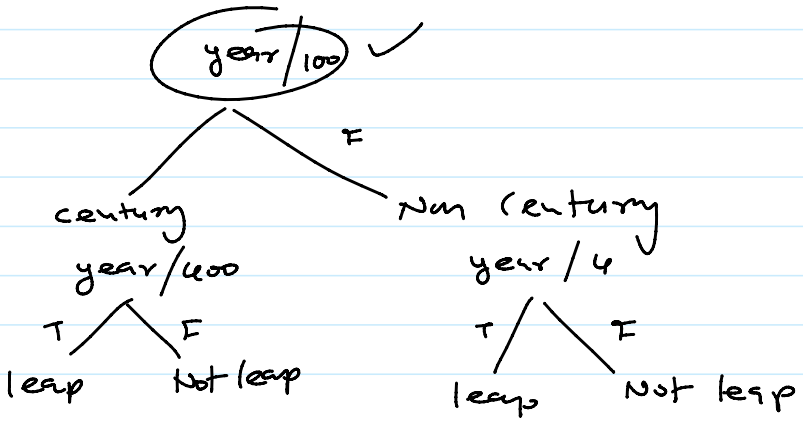
a, b, c, d

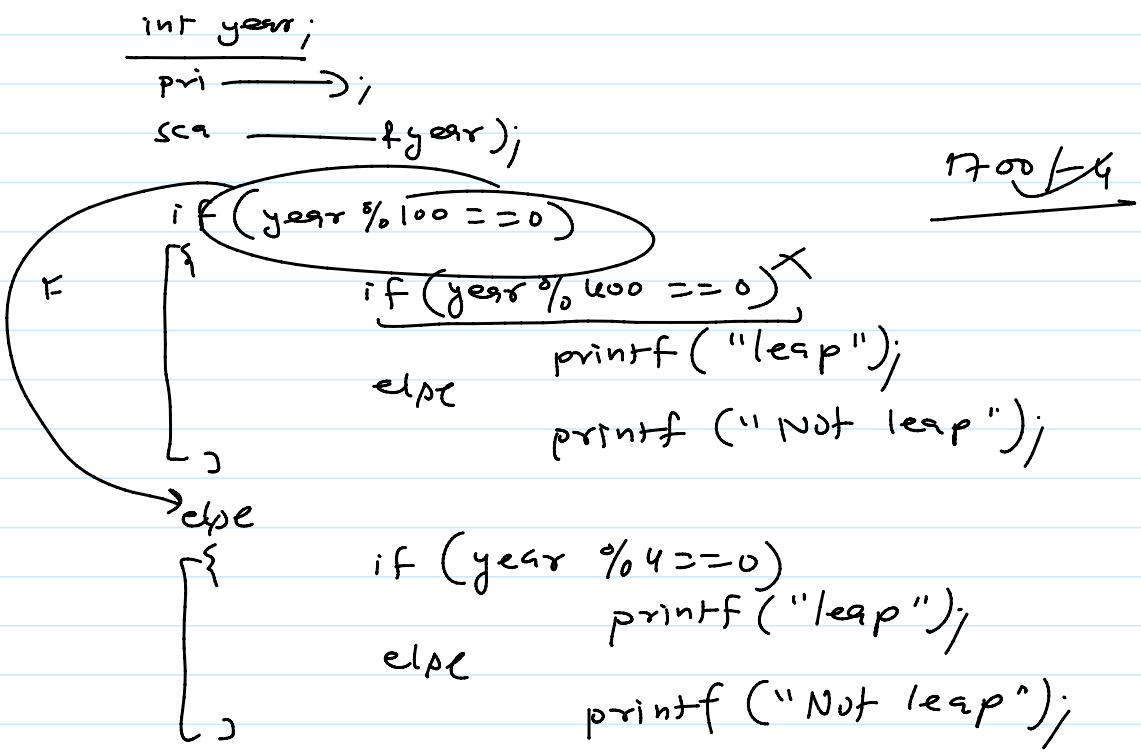


Q. leap year :-

1 year = ~~365.25 days~~ 365.248

	1 — 365	~~6~~ X	5.45.48
	2 — 365	6 X	
	3 — 365	6 X	
leap	4 — 365+1	6 + 18 → 24	
	⋮		
	96 — 365	6 + 18 → leap	
	97 — 365	6X	
	98 — 365	6X	
	99 — 365	6 X	18 hrs
~~leap~~	100 → 365	6 ~~+ 18~~ = 24	
	200 365	6 + 18	
	300 365	6 + ~~18~~	
leap	400 365	6 + ~~18~~ 18 = 24	





if else

```
if ( year %400 ==0 || year %100 != 0 && year %4==0)
                    printf ("lear year");
else
                    printf ("Not leap year");
```

1700/400 ✗
2012/100 ✗ → /4 ✓
7575
1600
1200
2010
2016
2000
2024

year %400 ==0 || year %100 != 0 &&
year %4 == 0)