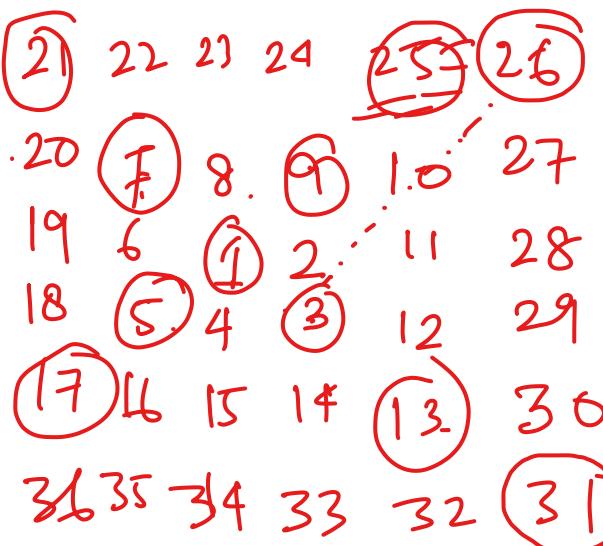


$\overbrace{DDRR}^{2N},$   
 $\overbrace{DRDR}^{\cdot},$   
 $\overbrace{DRRD}^{\cdot}$

exactly size  $2N$  of which there are

$\frac{N R's}{2N C_N}$  and  $N P's$

$$\boxed{\frac{(2N)!}{N! N!}} \quad \checkmark$$



Odd length square

$$(4n^2 - 6n + 6)$$

$$\left( \frac{2s}{3} \right) \left( 8s^2 + 15s + 3 \right) + 1$$

$$s = \frac{n-1}{2}$$

for  $s = 5$

101

$$\begin{array}{c} (13 \ 17 \ 21 \ 25) \\ (31 \ 37 \ 43 \ 49) \dots \\ \downarrow \qquad \qquad \qquad \downarrow \\ (3 \ 5 \ 7 \ 9) \end{array}$$

$$\begin{aligned} & (n^2 - 3n + 3) \\ & = 1^2 - 3 + 3 \\ & = 4 - 12 + 3 \\ & = \end{aligned}$$

$$\begin{aligned} & (3^2 - 3 \times 3 + 3) \\ & = 9 - 9 + 3 \\ & = 3 \\ & 9 - 2 \times 3 + 2 \\ & = 9 - 6 + 2 \\ & = 5 \\ & 9 - 2 \times 3 + 2 = 7 \end{aligned}$$

## shifting

3rd variable

```

int a, b, temp;
printf ("enter two no");
scanf ("i/d -f/d", &a, &b);
temp = a;
a = b;
b = temp;

```

printf (" value after shifting i/d -f/d", a, b);

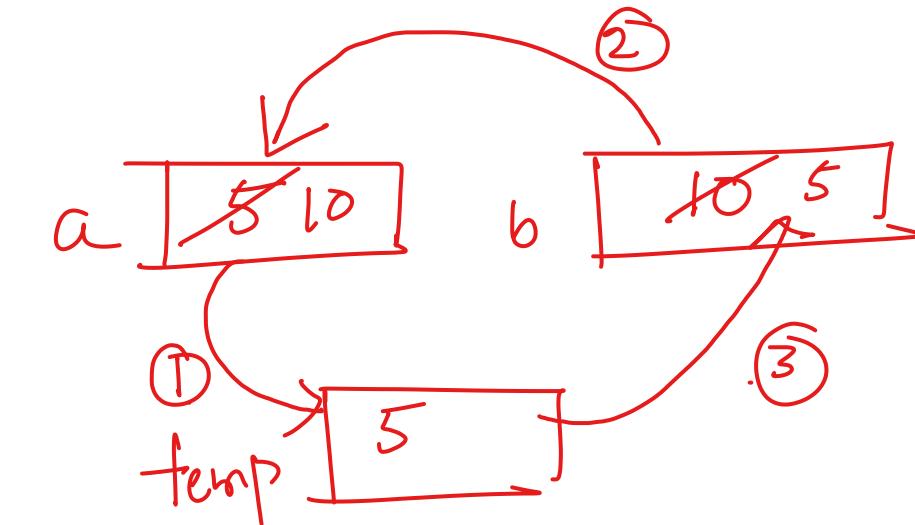
$$\left\{ \begin{array}{l} a = a + b; a = 5 + 10 = 15; a = 5 \quad b = 10 \\ b = a - b; b = 15 - 5 = 10; \\ a = a - b; a = 15 - 10 = 5 \end{array} \right.$$

bitwise  
operator

$$a = a \wedge b;$$

$$b = a \wedge b;$$

$$a = a \wedge b;$$



$$\underline{a=5}, \underline{b=3}$$

$$a = 101.$$

$$b = 011.$$

$$Y = \overline{A}B + A\bar{B}$$

$$\begin{array}{r} \overline{a} = 11 \cdot 0 \\ \overline{b} = 01 \cdot 1 \\ \hline b = 101 = \underline{\underline{5}} \end{array}$$

$a = 110$

|       |     |     |
|-------|-----|-----|
| $a =$ | 011 | = 3 |
|-------|-----|-----|

odd or even

{ if ( $n \% 2 == 0$ )  
    printf (" /d is even, n );  
else  
    printf (" /d is odd, n );

bit-wise    Operations

check    Odd & Even

## Switch Case

```
for (u=1; n<=5; n++)
```

```
    if (n == 3)
        break, continue;
```

```
    printf (" number = %.d", n);
```

```
)
```

n=1

n=2

n=1  
n=2  
n=4  
n=5

```
#include <stdio.h>
int main (void)
```

```
    int choice;
```

```
    printf (" enter your choice");
    scanf ("%d", &choice);

```

```
    switch (choice)
```

Second  
Third  
wrongchoice

choice : 2  
second.

```

    } case 1 : printf ("First\n");
    } case 2 : printf ("Second\n");
    } case 3 : printf ("Third\n");
    } default : printf ("wrong choice");
    } return 0;
}

```

int main()

```

    { int i=3;
    switch(i)
        { default : printf ("zero");
        case 1 : printf ("one");
        break;
        case 2 : printf ("two");
        break; case 3 : printf ("three");
        break;
    }
    return ;
}

```

Case 1 : printf ("First  
break");  
Case 2 : printf ("Second  
break");  
Case 3 : printf ("Third  
break");  
default : printf ("wrong choice");  
return 0;

O/p: Three

The default case can be placed anywhere inside the loop. It is executed only when all other cases does not match.

`printf (" -/x) -1<<4);`      O/p:-

↓                  ↓

Hexadecimal form      left shifting  
4 bits

-1 no. is stored in 2's complement

0000 0000 0000 0001

O/p:- FFF0

1111 1111 1111 1110  
+ 1

1111 1111 1111 1111 <<4

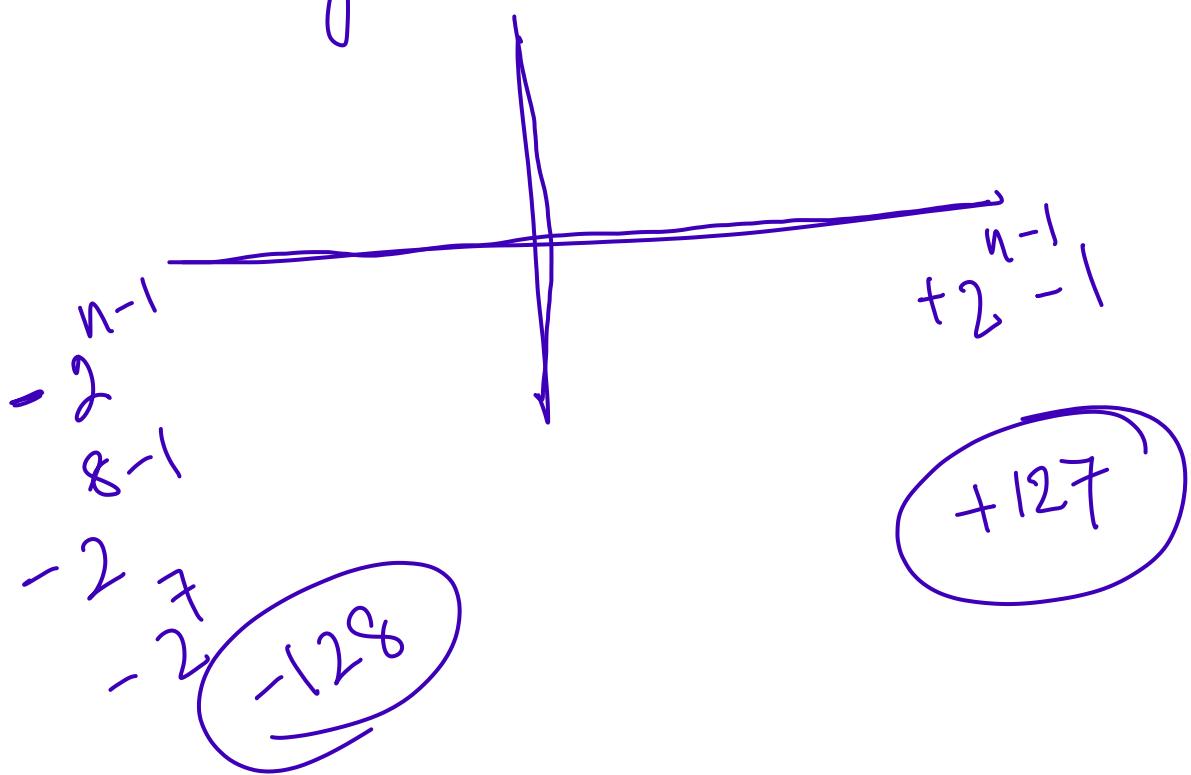
1111 1111 1111 0000  
F F F O

factorial

$$n! = n(n-1)(n-2) \dots$$

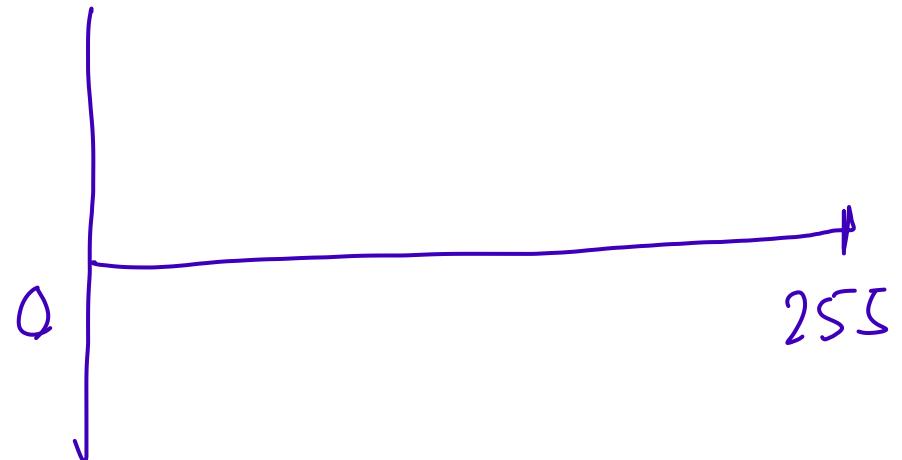
char a;

1 byte = 8 bits



int a;  
signed

unsigned char a;



int

4 bytes -

unsigned int

~~8 bytes~~

long unsigned int

$-32768$  to  $+32767$

$65535$

```

#include <stdio.h>
int main (void)
{
    int n, i = 1;
    long unsigned int f = 1;
    printf (" Given an Integer");
    scanf ("%d", &n);
    if (n < 0)
        printf (" Invalid input, must be positive");
    exit (0);
}

while (i <= h)
{
    f *= i;
    i++;
}
printf (" Factorial of %d is %c", n, f);
}

```

exit(0) → both will terminate the program

exit(1) : normal termination of the program

exit(1) Program is terminated because of an error

h = 3  
 while ( $i \leq 3$ )  
 {  
 f = f \* i = 1 \* 1 = 1  
 i = 2  
 }

while ( $i \leq 3$ )  
 {  
 f = 1 \* 2 = 2  
 i = 3  
 }

while ( $i < 3$ ) X

$$\begin{aligned} f &= 2 * 3 = 6 \\ i &= 4 \end{aligned}$$











