Mon to Wed → 9-12 Class
Thu → OFF ——————→ OFF from teacher
Fri → 9-12 Class
Sat ⇒ Tuts 9
Sun → Test discussion + doubts + topic continuation

① Digit Frequency, given number n & digit d, find the frequency of given digit d in the given number

Eg. 9 9 4 5 4 3 2 3 4 , 4

Approach, check for each digit from last and if equal to d then freq ++;

CODE

```
int freq = 0;
while (n > 0) {
    if (n % 10 == d)
        freq++;
    n = n / 10;
}
```

☆ Functions: A block of code that can be called upon to perform a particular task. good alternative to repetetive code.

Eg. for factorial in calculation of ${}^{n}C_{r} = \frac{n!}{r! \times (n-r)!}$

function to calculate factorial

```
int fact (int num) {
    int ans = 1;
```

```
    for (int i = 2; i <= n; i++)
        ans = ans * i;
    return ans;
}
```

Now, to calculate $^{n}C_{r}$ → C = fact(n)/(fact(r) * fact(n-r))

② Decimal to any base

Eg: $(634)_{10} \rightarrow (\quad)_{8}$

Step ①, divide 634 by 8

dest. base	Quotient	Remainder
8	634	2
8	79	7
8	9	1
	1	

Step ② answer is the remainders in backwards order

$(1\ 1\ 7\ 2)_{8} = (6\ 3\ 4)_{10}$

$(512 + 64 + 56 + 2)_{10}$

$= (6\ 3\ 4)_{10}$ ✓✓

∴ our answer is reverse of the remainders obtained by dividing the number by the new base

☆ Imp. terms Process, Stack, Heap, Procedural Programming

CODE:

```
int ans = 0, pow10 = 1;
while (n > 0) {
```

```
    ans = ans + (n%b) * pow10;
    n = n/b;
    pow10 = 10 * pow10;
}
return ans;
```

n%b → gives remainder of n

(n%b) X pow10 → gives correct place to the face value of n%b

③ any base to decimal

Eg: 1 1 1 0 0 1 , b = 2

2^5 2^4 2^3 2^2 2^1 2^0

For each digit (n%10) with condition n = n/10 each time

ans = ans + (n%10) X powb

& powb = b X powb

CODE

```
int ans = 0, powb = 1;
while (n > 0) {
    ans = ans + (n%10) X powb;
    powb = b X powb;
    n = n/10;
}
return ans;
```

④ any base to any base

Eg: 1 1 1 0 0 1 b_1 = 2 b_2 = 3.

2^5 2^4 2^3 2^2 2^1 2^0

↓

convert to decimal ③

↓

convert to base b2 ②

(5) Any base addition.

Eg: b=8 n1 = 7 7 7 n2 = 1 1 0 0 0

8^2 8^1 8^0 + 8^0 = 8^3 8^2 8^1 8^0

CODE

```
int ans=0, pow10=0, carry=0;
while(n1>0 || n2>0 || carry>0){
    int res = (n1%10)+(n2%10) + carry;
    ans = ans + (res%b) x pow10;
    pow10 = 10 x pow10;
    n1 = n1/10; n2 = n2/10;
    carry = res/b;
}
return ans;
```