

## Basic maths concept

① extraction of digits

$$N = 7889$$

$$\begin{aligned} N/10 &\rightarrow 7889 \div 10 = 9 \\ N/10 &\rightarrow 788 \div 10 = 8 \\ N/10 &\rightarrow 78 \div 10 = 8 \\ N/10 &\rightarrow 7 \div 10 = 7 \\ N/10 &\rightarrow 0 \div 10 = 0 \end{aligned}$$

$$\text{int } N = 7889;$$

while ( $N > 0$ ) {

↓

$$\text{last digit} = N \% 10;$$

$$N = N / 10;$$

}

funcn :

```
int count (int n) {
```

```
    int cnt = 0;
```

```
    while (n > 0) {
```

```
        int last_digit = n % 10;
```

```
        Cnt = Cnt + 1;
```

```
        n = n / 10;
```

}

```
    return Cnt;
```

}

if no. of iterations is  
based on divisibility,  
then we use this  
formula

Time complexity =  $O(\log_{10}(N))$

### ② Reverse a number

Consider previous example  
 $(N = 9887)$

$$9887 \div 10 = 7$$

$$988 \div 10 = 8$$

$$98 \div 10 = 8$$

$$9 \div 10 = 9$$

funcn:

```
int rev_no (int n){
```

```
    int rev = 0;
```

```
    while (n > 0){
```

last-digit =  $n \div 10$ ;

$n = n / 10$ ;

$rev = (rev \times 10) + last-digit;$

}

return rev;

}

### ③ Palindrome number

$\hookrightarrow$  No. of which reverse is same as original one

Ex: 121

logic:

- 1) find reverse of no.
- 2) Compare it with original value
- 3) if both same, then true  
 if not then false

int main(){

funcn:

```
int n;
```

```
cin >> n;
```

```
int rev_num = 0;
```

```
int dup = n;
```

```
while (n > 0){
```

int ld =  $n \div 10$ ;

$rev\_num = (rev\_num \times 10) + ld;$

$n = n / 10$ ;

}

if (dup == rev\_num){

cout << "true";

else{

cout << "false";

#### ② Armstrong number

N = 371

If sum of cube of every digit is equal to no. itself  
then it is an Armstrong number.

N = 371

$$3^3 + 7^3 + 1^3 = 371$$

Paged No. \_\_\_\_\_

Date \_\_\_\_\_

#### ③ Armstrong number

1) Extract all digit

Logic: 2) Cube of all digit and sum them  
3) If equal then true if not then false

funcn:

int dup = N;

int sum = 0;

while (N > 0) {

    last\_digit = N % 10;

    sum = sum + (last\_digit \* last\_digit \* last\_digit);

    N = N / 10;

}

if (sum == dup) {

    cout << "true";

} else {

    cout << "false";

}

3

#### ④ Print all divisors

↳ Should lie b/w 1 to N

(i) Remainder should be zero

Logic:

funcn:

void print\_divisors(int n) {

    for (int i = 1; i <= n; i++) {

        if (n % i == 0) {

            cout << i << " ";

}

(5) Prime number check  
exactly 2 factors (1 and itself)

funcn:

```
for (int i = 1; i * i <= n; i++) {
    if (n / i == 0) {
        cnt++;
    }
    if ((cnt * i) != i) cnt++;
}
if (cnt == 2) cout << "true";
else cout << "false";
```

(6) GCD / HCF

funcn:

int n1;

int n2;

```
for (i = min(n1, n2); i >= 1; i--) {
    if (n1 % i == 0 && n2 % i == 0) {
        cout << i;
        break;
}
```

~~break~~

}

Time Complexity =  $O(\min(n_1, n_2))$