Notes on Problems :-

Basic Maths :-

Problem 1: Given an integer N, write a program to count the number of digits in N.

Solution :

Approch 1 : 1. store the integer in variable x. and initialize a counter variable to count the number of digits.

2. When we divide x by Y it will result in an integer (given both the variables are integer). For example, 156/10 will result in 15 . similarly 5/10 will result in 0.

3. Using for loop and above observation keep on dividing x by 10. (x is input integer) and increment the count in every iteration when x equals 0 terminate the count and the count will have number of digits in N. O(N)

Approach 2 :

1) convert the integer N to string.
2) return the size of string. O(1)

Approach 3:

Use logarithm base 10 to count the number of digits. As

The number of digits in an integer

= the upper bound of $\log_{10}(n)$.

Problem 2: Reverse a given number and print it.

Solution:

Approach 1:

1. convert the number into string.
2. Reverse the string.

Approach 2:

The idea is to extract digits from the end of the given number and create a new number in reverse order.

How to extract digits from the end of a number?

⇒ To extract the last digit, If you divide a number by 10, then the reminder will be last digit.

We can use the modulo (%) operator to do this,.

Ex. $123\%10 = 3$.

and to reduce the number by 1 digit, simply divide the number by 10.

To create a number from extracted digits:

⇒ The idea is to start with 0, and for every digit, multiply the number generated so far by 10, and add the digit to it.

Algorithm:

- Run a while loop until the given number N is equal to zero (N!=0).
- Initialize a variable reverse=0;
- now in each step take the reminder of the given number N and store it as a variable digit,
  digit= N%10;
- Also divide the number by 10, N=N/10;
- in each step, variable reverse get updated as reverse= reverse*10 + digit.

Problem 3: Given a number, check if it is a pallindrome.

Solution:

1) Reverse the number using algorithm given in Problem 2.
2) check if original & reversed numbers are equal or not.

Problem 4: Find the GCD of two numbers.
Greatest common divisor (GCD) and Highest common Factor (HCF) are same.

Solution:

Approach 1: Simply Traverse From 1 to min(a,b) and check for every number.

- Traverse from 1 to min(a,b).
- And check if i can divide by both a and b. If yes store it in the answer.
- Find the maximum valve of i which divides both a and b.

Approach 2:

Using Eucleadian's theorem.

Intuition: GCD is the greatest number which can divide both a and b. If a number can divide both a and b. it should also divide both (a-b) and b as well.

$$gcd(a,b) = gcd(b, a\%b) \qquad a \geq b$$

Appraach:

- Recursively call gcd(b, a%b) function till the base condition is hit.

- b=0 is the base condition. When base condition is hit return a, as gcd(a,0) equal to a.

Problem 5: Given a number, check if it is a Armstrong number or not.

Armstrong numbers:

Armstrong numbers are the numbers having the sum of digits (individual digits of a number) raised to power no. of digits (in a number) is equal to a given number.

e.g. $1^3 + 5^3 + 3^3 = 153$

Number of digits d=3

$1^4 + 6^4 + 3^4 + 4^4 = 1634$.

Number of digits, d= 4.

Solution:

1. count the number of digits in a number.
2. initialize variable sum=0.
3. extract each digit, caclulate digit$^{\text{no. of digits}}$. and add it to sum.
4. Repeat step 3 for each digit.
5. check if sum is equal to original number.

Problem 6: Print all divisors of a given number.

Given a number, print all the divisors of the number. A divisor is a number that gives the remainder as zero when divided.

Solution:

Approach 1: Brute Force.

start from 1 and check each number until n. if i divides n add it to list.

Approach 2:

If we take a closer look, we can notice that quotient of a division by one of the divisors is actually another divisor. Like, 4 divides 36. The quotient is 9, and 9 also divides 36.

Another intuition is that root of a number actually acts as a splitting part of all the divisors of a number.

Also, the quotient of a division by any divisor gives an equivalent divisor on the other side. like 4 gives 9 while dividing 36.

- From the intuition, we can come to conclusion that we dont need to traverse all the candidates

and if we found a single divisor, we can also find another divisor (Here, we need to be careful, if the given number is a perfect square, like 36, 6 also give 6 as a quotient. This is a corner case.)

- By keeping this in mind, it is enough if we traverse up to the root of a number. Whenever we find the divisor, we print it and also print the quotient. If the quotient is the same, we ignore it. Because the root of a perfect square will give the same quotient as itself.
- The quotients are the numbers that are on the other side of the root. so, it's okay if we stop traversing at the root.

Problem 7: check if a number is prime or not.

Given a number, check wheather it is prime or not. A prime number is a natural number that is only divisible by 1 and itself.

Solution:

Brute Force:

check if number is divided by 2.

from 3 to n, check if it is divisible by any odd number.

We can optimise above solution, by only checking till the square root of a given number.