1. WAP to accept a number and display all the digits from left to right without reversing the number.

Ex - 1231

1
2
3
1

2. WAP to accept a number and display the prime decomposition of the number

75 → 3 x 5 x 5

```
void main(int num)
{  int d, count = 0, copy = num ;
  while (copy > 0)
  { count++;
    num /= 10;
  }
  while (count > 0)
{     d =  num / (int) Math.pow(10, count-1);
 System.out.println(d);
 num = num % (int) Math.pow(10, count-1);
 count-- ;
 }
  }
```

1234 → 4 3 2 1

```
void main (int num)
{   int i;
    for (i= 2; i<=num ; i++)
        if (num %i==0)
        {   printf (i);
            num = num / i;
            i = i - 1;
        }
}
```

i → 2, 3, 4, 5, 6, 7

2, 3

2 2 2

18.09

1) WAP to accept a number and swap the 1st and last digit and display the number eg. → 1234 → 4231

2) WAP to accept a number and check the number is odd or even without using % and / and if.

3) WAP to find the quotient and remainder of a division without using % and /

4. WAP to accept a number and arrange the digits in decending order and display the number. 5629 ⇒ 9652

5) WAP to Convert a Binery number to decimal number 110 ⇒ 6

$1\times 2^2 + 1\times 2^1 + 0\times 2^0$

192, 384, 576 — 184, 392

K Kx2 Kx3

Third num.

void main(int num)

{ int count=0, d1, d2, copy = num

d2 = copy % 10;

while (copy > 0)

{ d1 = copy % 10

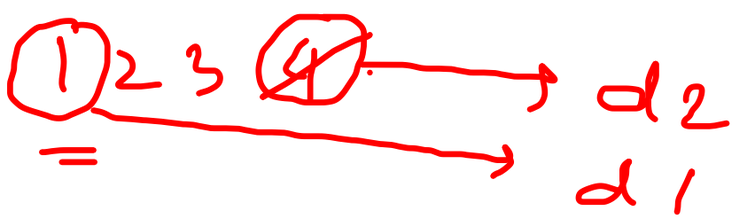
count++; copy /= 10;

}

int newnum = num / 10;

newnum = newnum * (int) Math.pow(10, count-2);

newnum = newnum * 10 + d1



d * 1000 + 231.

newnum = d2 * (int) Math.pow(10, count-1) + newnum;

S.O.P(newnum);

}

```
void main (int num, int div)
{
    int quo = 0
  while (num >= div)
  {
      quo++;
      num = num - div;
  }
    S.O.P ("Quotient = " + quo);
S.O.println ("Remainder = " + num);
}
```

6, 2

quo = 1, num = 4

quo = 2 num = 2

quo = 3 num = 0

5 >= 3

1

2

1

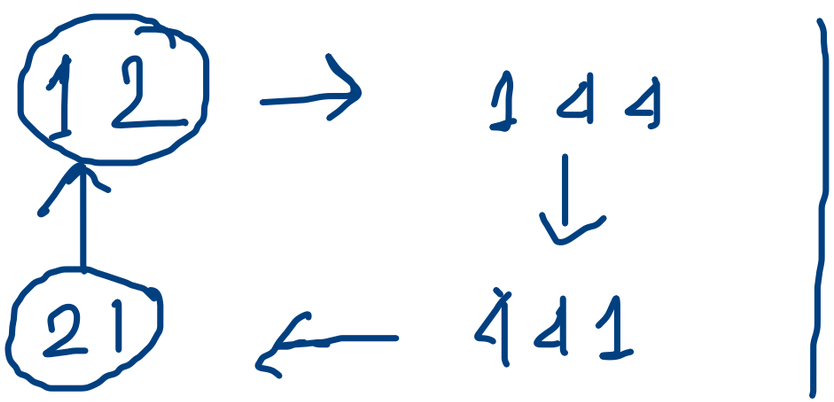
→ 2

```
void main (int rev)
{  int i, d, revrev=0, copy,      ;
  for (i= 9; i >= 0; i--)
    {   copy = rev ;                    S.o print(revrev);
        while (copy>0)                          }
    { d = copy.10;
      if( d == i )
          revrev = revrev *10 + d;
    } copy/=10;
}
```

```
void main( int num)
{
   int i      ;
  for (i= num; i>=20; i-=2)
                ;
  s.o p( i==0 ? "even" : "odd" ) ;
}
```

25.09.

1) Prog to display all the two digit number without repetition which has the following properties



2) Prog to accept a number and check the number is circular prime or not.

Example: 313 → prime

133 → prime

331 → prime

do --- while Loop.

Syntax :

```
do
{
  ---
  ---
} while (condition);
```

↓

```
int n = 5;
do
{
  s.o p(n);
  n--;
} while(n>0);
```

↓

(*) This is exit control loop structure which means there is no test expression to enter in the loop and the condition is given at the end to check the exit condition.

(7) Everytime there will be at least one iteration

eg → int n = 0
do { S.O P(n);
} while (n > 0);

int n = 0
while (n > 0)
S.O P(n);

Convert the following to while, for

int n = 10;
do { n = n - 2;
S.O P(n+2);
} while(n > 3);

```
int n=10;
while(n>3)
{
  n=n-2;
  s.o.p(n);
}
```

```
int n;
for(n=10; n>3;)
{
  n=n-2;
  s.o.p(n);
}
```

(1) WAP to display ~~all~~ ~~the~~ whether a number is triangular number or not.

Eg → 6 → 1+2+3

10 → 1+2+3+4

② WAP to display all the two digits triangular number.

③ WAP to accept a number and check whether it is found in fibonacci series or not.