what are the doubts in single loop.

Palindrome

Armstrong

Table

Prime number

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public void checkPrime(int number) // number is storing 43

{

int half=number/2; // 43/2---21

int sahil=0;

for(int i=2; i<=half;i=i+1) // 2 to 21 i-----21

{

if(number%i==0) // 43%21--1

{

syso("its not a prime number" +number);

sahil=1;

break; // for moving out of the loop

}

}

if(sahil==0)

{

syso("its a prime number"+number);

}

}

main method

obj.checkPrime(43);

which is the biggest number that can divide 44 except 44

22 is the biggest number that can divide 44, 22 is the half of the number.

which is the smallest number that can divide 44 except 1

2 is the smallest number that can divide 44.

48------after 24 no number can divide 48 and remainder is 0

so execute the loop till half of the number

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

what is palindrome number

reversal of the number is equal to the original number

public void checkPalindrome(int number)

{

int original\_number=number; // 141

int sum=0;

while(number>0) 141>0 14>0 1>0 0>0

{

int r=number%10; 141%10----1 14%10-----4 1%10----1

sum=sum\*10+r; 0\*10+1--------1 1\*10+4----14 14\*10+1----141

number=number/10; 141/10----14 14/10--------1 1/10-------0

}

if(original\_number==sum)

{

syso("its a palindrome number"+original\_number);

}

else

{

syso("not a palindrome number"+original\_number);

}

main method-----------obj.checkPalindrome(141);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

armstrong number==========sum of each digit cube

153-----------------1+125+27-------153

logic will be same --------------------------------as we did for palindrome

change will be

sum=sum+r\*r\*r; // sum=sum\*10+r;

if(number>0)&&(number<=999) 884>0 884<999

{

while(number>0)

{

int r=number%10;

sum=sum+r\*r\*r;

number=number/10;

}

}

else if(number>999)&&(number<=1000)

{

int r=number%10;

sum=sum+r\*r\*r\*r;

number=number/10;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public void printtable()

{

int number=sc.nextInt(); 5

for(int i=1;i<=10;i=i+1)

{

syso("number "+number+"\*"+i+"="+number\*i);

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

we need two loops one is for row and one is for column

how many rows we have---------------5

first row---------------------------one star

second row----------------------two star

third row--------------------------three star

fourth row-----------------------four star

fifth row-------------------------five star

double loop

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

1

12

123

1234

12345

1

2 3

4 5 6

7 8 9 10

three loops required ?

one for rows

second for printing space in the column

third for printing \* in the column

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

outer loop

inner loop for printing space

inner loop for printing \*

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*