/Q1.a number is even or not, including +ve,-ve and 0\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int isEven(int input1){

// Read only region end

// Write code here...

if(input1%2==0)

return 2;

return 1;

}

}

--------------------------------------------------------------

Q2. a number is odd or not, including +ve,-ve and 0.

ans2. same as above.

--------------------------------------------------------------

Q3. find then last digit of a number.

Eg 197=7

-197=7

Int num 🡪;

num = math.abs(num);

l = num%10;

SOP(l); SOP(num%10);

ans3. use Math.abs();

--------------------------------------------------------------

Q4. find the 2nd last digit of a number.

eg 197 or -197=9

19 = 1

5=0

C=0;

ans4. while(c!=2) //num = 5;

{

d=num%10;

c++;

num=num/10;

if(num==0)

break;

}

if(c==1)

print(0);

else

print(d);

--------------------------------------------------------------

Q5. Find the sum of the last digits of two numbers;

Eg n1 = 12, n2 = 13

Sum = 2+3 = 5

ans5. sum=Math.abs(num1%10)+Math.abs(num2%10);

//Q6. Input two nos and find if 1st number is a multiple of 2nd or not.

Eg n1 = 18 n2 = 6

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int isMultiple(int input1,int input2){

// Read only region end

// Write code here...

if((input1==0)||(input2==0))

return 3; // if any 1 is 0

else

{

if(input1%input2==0)

return 2;

return 1;

}

}

}

----------------------------------------------------------------

//Q7. Out of 5 numbers, how many are even?

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int countEvens(int input1,int input2,int input3,int input4,int input5){

// Read only region end

// Write code here...

int c = 0;

if(input1%2==0)

c++;

if(input2%2==0)

c++;

if(input3%2==0)

c++;

if(input4%2==0)

c++;

if(input5%2==0)

c++;

return c;

}

}

----------------------------------------------------------------

/\*Q8. Count no of odd num from given 5 nums\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int countEvens(int input1,int input2,int input3,int input4,int input5){

// Read only region end

// Write code here...

int c = 0;

if(input1%2!=0)

c++;

if(input2%2!=0)

c++;

if(input3%2!=0)

c++;

if(input4%2!=0)

c++;

if(input5%2!=0)

c++;

return c;

}

}

----------------------------------------------------------------

/\*Q9. String given even or odd. count according to it\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int countEvensOdds(int input1,int input2,int input3,int input4,int input5,String input6){

// Read only region end

// Write code here...

//throw new UnsupportedOperationException("countEvensOdds(int input1,int input2,int input3,int input4,int input5,String input6)");

int even\_count = 0,odd\_count = 0;

if(input6.equalsIgnoreCase("Even"))

{

if(input1%2==0)

even\_count++;

if(input2%2==0)

even\_count++;

if(input3%2==0)

even\_count++;

if(input4%2==0)

even\_count++;

if(input5%2==0)

even\_count++;

return even\_count;

}

else

{

if(input1%2!=0)

odd\_count++;

if(input2%2!=0)

odd\_count++;

if(input3%2!=0)

odd\_count++;

if(input4%2!=0)

odd\_count++;

if(input5%2!=0)

odd\_count++;

return odd\_count;

}

}

}

----------------------------------------------------------------

/\*Q10. a number is prime or not\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int isPrime(int input1){

// Read only region end

// Write code here...

int i,c = 0;

for(i=1;i<=input1;i++)

{

if(input1%i==0)

c++;

}

if(c==2)

return 2;

return 1;

}

}

----------------------------------------------------------------

/\*Q11. find the factorial of a number\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int nFactorial(int input1){

// Read only region end

// Write code here...

int i,f = 1;

for(i=1;i<=input1;i++)

f = f\*i;

return f;

}

}

----------------------------------------------------------------

/\*Q12. Find the nth fibonacci number\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public long nthFibonacci(int input1){

// Read only region end

// Write code here...

int a=0,b=1,c=0,i;

if(input1==1)

return 0;

else if(input1==1)

return 1;

else

{

for(i=3;i<=input1;i++)

{

c=a+b;

a=b;

b=c;

}

return c;

}

}

}

----------------------------------------------------------------

/\*Q13. Find the nth prime number\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int NthPrime(int input1){

// Read only region end

// Write code here...

int count = 0,i,c,num=2;

while(count!=input1)

{

while(true)

{

c=0;

for(i=1;i<=num;i++)

{

if(num%i==0)

c++;

}

if(c==2)

count++;

if(count==input1)

break;

num++;

}

if(count==input1)

break;

}

return num;

}

}

----------------------------------------------------------------

/\*Q14. find prime nums in a given range\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int countPrimesInRange(int input1,int input2){

// Read only region end

// Write code here...

int i,c,j,count = 0;

for(i=input1;i<=input2;i++)

{

c=0;

for(j=1;j<=i;j++)

{

if(i%j==0)

c++;

}

if(c==2)

count++;

}

return count;

}

}

----------------------------------------------------------------

/\*Q15. Count no of digits present in a number\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int allDigitsCount(int input1){

// Read only region end

// Write code here...

int count=0,d;

while(input1!=0)

{

count++;

input1=input1/10;

}

return count;

}

}

----------------------------------------------------------------

/\*Q16. count the unique digits in a number\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int uniqueDigitsCount(int input1){

// Read only region end

// Write code here...

int dig[]=new int[5];

int i,k=0,count=0,j;

while(input1!=0)

{

dig[k]=input1%10;

k++;

input1=input1/10;

}

for(i=0;i<k-1;i++)

{

for(j=i+1;j<k;j++)

{

if(dig[i]==dig[j])

dig[j]=-1;

}

}

for(i=0;i<k;i++)

{

if(dig[i]!=-1)

count++;

}

return count;

}

}

----------------------------------------------------------------

/\*Q17. count the non repeated digits\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int nonRepeatDigitsCount(int input1){

// Read only region end

// Write code here...

int dig[]=new int[5];

int i,k=0,count=0,j,flag;

while(input1!=0)

{

dig[k]=input1%10;

k++;

input1=input1/10;

}

for(i=0;i<k-1;i++)

{

flag=0;

for(j=i+1;j<k;j++)

{

if(dig[i]==dig[j])

{

flag=1;

dig[j]=-1;

}

if(flag==1)

dig[i]=-1;

}

}

for(i=0;i<k;i++)

{

if(dig[i]!=-1)

count++;

}

return count;

}

}

OR

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int nonRepeatDigitsCount(int input1){

// Read only region end

// Write code here...

int dig[]=new int[5];

int freq[]=new int[5];

int i,c,k=0,count=0,j;

while(input1!=0)

{

dig[k]=input1%10;

k++;

input1=input1/10;

}

for(i=0;i<k;i++)

{

c=0;

for(j=0;j<k;j++)

{

if(dig[i]==dig[j])

{

c++;

}

}

freq[i]=c;

}

for(i=0;i<k;i++)

{

if(freq[i]==1)

count++;

}

return count;

}

}

\*\*

/\*Q18. Find sum of digits till you get the sum in the range 1 to 9\*/

/\*for –ve num, sum should be –ve\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int digitSum(int input1){

// Read only region end

// Write code here...

int sum=0,flag=0;

if(input1<0)

flag=1;

input1=Math.abs(input1);

while(input1>9)

{

sum=0;

while(input1!=0)

{

sum=sum+(input1%10);

input1=input1/10;

}

input1=sum;

}

if(flag==0)

return sum;

else

return (-1\*sum);

}

}

-----------------------------------------------------------------/\*Q19. find sum of digits that are even only\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int EvenDigitsSum(int input1){

// Read only region end

// Write code here...

int sum=0,d;

while(input1!=0)

{

d=input1%10;

if(d%2==0)

sum=sum+d;

input1=input1/10;

}

return sum;

}

}

-----------------------------------------------------------------

/\*Q20. find sum of digits that are odd only\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int OddDigitsSum(int input1){

// Read only region end

// Write code here...

int sum=0,d;

while(input1!=0)

{

d=input1%10;

if(d%2!=0)

sum=sum+d;

input1=input1/10;

}

return sum;

}

}

/\*Q21. Find sum of digits according to input2\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int EvenOddDigitsSum(int input1,String input2){

// Read only region end

// Write code here...

int sum=0,d;

if(input2.equalsIgnoreCase("even"))

{

while(input1!=0)

{

d=input1%10;

if(d%2==0)

sum=sum+d;

input1=input1/10;

}

return sum;

}

else

{

while(input1!=0)

{

d=input1%10;

if(d%2!=0)

sum=sum+d;

input1=input1/10;

}

return sum;

}

}

}

-----------------------------------------------------------------

/\*Q22. Find if a number is a palindron or not\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int isPalinNum(int input1){

// Read only region end

// Write code here...

int n,r=0,d;

n=input1;

while(input1!=0)

{

d=input1%10;

r=(r\*10)+d;

input1=input1/10;

}

if(r==n)

return 2;

else

return 1;

}

}

/\*Q23. find if palindrome no is possible or not by rearranging the digits\*/

import java.util.\*;

class anu

{

public static void main(String args[])

{

Scanner sc=new Scanner (System.in);

int num,k=0,i,j,c=0;

int a[]=new int[5];

int freq[]=new int[5];

System.out.println("Enter a number");

num =sc.nextInt();

//storing all the digits in an array

while(num!=0)

{

a[k]=num%10;

k++;

num=num/10;

}

for(i=0;i<k-1;i++)

{

c=0;

for(j=0;j<k;j++)

{

if(a[i]==a[j])

c++;

}

freq[i]=c;

}

c=0;

for(i=0;i<k;i++)

{

if(freq[i]%2!=0)

c++;

}

if(c>=2)

System.out.println("Palindrome not possible");

else

System.out.println("Palindrome possible");

}

}

------------------------------------------------------------------------------------------- - - - - ----- - - - - - - - - - - - - - - - - - -

/\*Q24. Generate PIN of 4 digit. Given three 3-digit number input.

Unit dig = min of all unit digit.

Ten dig = min of all ten digit.

Hund dig= min of all hund digit.

thous dig= max all digits.\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int createPIN(int input1,int input2,int input3){

// Read only region end

// Write code here...

int u,t,h,th,PIN,i,a=0,b=0,c=0,m1,m2,m3;

int unit[]=new int[3];

int ten[]=new int[3];

int hun[]=new int[3];

while(input1!=0)

{

unit[a]=input1%10;

a++;

input1=input1/10;

}

while(input2!=0)

{

ten[b]=input2%10;

b++;

input2=input2/10;

}

while(input3!=0)

{

hun[c]=input3%10;

c++;

input3=input3/10;

}

if((unit[0]<ten[0])&&(unit[0]<hun[0]))

u=unit[0];

else if((ten[0]<unit[0])&&(ten[0]<hun[0]))

u=ten[0];

else

u=hun[0];

if((unit[1]<ten[1])&&(unit[1]<hun[1]))

t=unit[1];

else if((ten[1]<unit[1])&&(ten[1]<hun[1]))

t=ten[1];

else

t=hun[1];

if((unit[2]<ten[2])&&(unit[2]<hun[2]))

h=unit[2];

else if((ten[2]<unit[2])&&(ten[2]<hun[2]))

h=ten[2];

else

h=hun[2];

m1=unit[0];

m2=ten[0];

m3=hun[0];

for(i=0;i<3;i++)

{

if(unit[i]>m1)

m1=unit[i];

if(ten[i]>m2)

m2=ten[i];

if(hun[i]>m3)

m3=hun[i];

}

if((m1>m2)&&(m1>m3))

th=m1;

else if((m2>m1)&&(m2>m3))

th=m2;

else

th=m3;

PIN=(th\*1000)+(h\*100)+(t\*10)+u;

return PIN;

}

}

/\*Q25. Do sum for the pattern 1 where input1=no. of rows.

2 2

3 3 3

input2=starting value, input3=incremeting weight value.\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int totalHillWeight(int input1,int input2,int input3){

// Read only region end

// Write code here...

int i,c=input2,sum=0,w=input3,j;

for(i=1;i<=input1;i++)

{

for(j=1;j<=i;j++)

sum=sum+c;

c=c+w;

}

return sum;

}

}

/\*Q26. Return the 2nd word of a sentence in upper case. if single word given then return "LESS".\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public String secondWordUpperCase(String input1){

// Read only region end

// Write code here...

input1=input1.trim();

int i,l,f=0,j=0,p=0,q;

char ch;

String s1="";

char b[];

l=input1.length();

b=new char[l];

//input1=input1+" ";

for(i=0;i<l;i++)

{

ch=input1.charAt(i);

b[j]=ch;

j++;

}

for(i=0;i<j;i++)

{

if(b[i]==' ')

{

p=i;

break;

}

}

//case of single word.

if(p==0)

f=1;

q=p;

for(i=p+1;i<j;i++)

{

if(b[i]==' ')

{

q=i;

break;

}

}

if(f==1)

{

s1="LESS";

return s1;

}

else

{

if(q==p)

{

for(i=p+1;i<j;i++)

s1=s1+b[i];

s1=s1.toUpperCase();

return s1;

}

else

{

for(i=p+1;i<q;i++)

s1=s1+b[i];

s1=s1.toUpperCase();

return s1;

}

}

}

}

/\*Q27. Check for palindrome string. if true return 2.\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int isPalindrome(String input1){

// Read only region end

// Write code here...

char a[];

char ch;

int i,l,j=0,k,f=0;

l=input1.length();

a=new char[l];

input1=input1.toUpperCase();

for(i=0;i<l;i++)

{

ch=input1.charAt(i);

a[j]=ch;

j++;

}

for(i=0,k=j-1;i<k;i++,k--)

{

if(a[i]!=a[k])

{

f=1;

break;

}

}

if(f==1)

return 1;

else

return 2;

}

}

/\*Q28. Find the weight of string according to their position in the alphabet list. ignore space and other characters.

Input2 if 0, then vowels not be added.

input2 if 1, then vowels to be added.\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int weightOfString(String input1,int input2){

// Read only region end

// Write code here...

int i,l,sum=0,j;

char ch;

char a[]={' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};

l=input1.length();

input1=input1.toLowerCase();

if(input2==0)

{

//ignore

for(i=0;i<l;i++)

{

ch=input1.charAt(i);

for(j=1;j<=26;j++)

{

if(ch==a[j])

{

if((ch=='a')||(ch=='e')||(ch=='i')||(ch=='o')||(ch=='u'))

sum=sum+0;

else

sum=sum+j;

}

else

sum=sum+0;

}

}

return sum;

}

else

{

//accept

for(i=0;i<l;i++)

{

ch=input1.charAt(i);

for(j=1;j<=26;j++)

{

if(ch==a[j])

{

sum=sum+j;

}

else

sum=sum+0;

}

}

return sum;

}

}

}

/\*Q29. Find the digit with the highest frequency out of 4 given input. If 2 or more digits share the same highest freq, then return the maximum out of them.\*/

import java.io.\*;

import java.util.\*;

// Read only region start

class UserMainCode

{

public int MostFrequentDigit(int input1,int input2,int input3,int input4){

// Read only region end

// Write code here...

int n[]=new int[20];

int f[]=new int[20];

int s[]=new int[20];

int i,k=0,p=0,j;

while(input1!=0)

{

n[p]=input1%10;

p++;

input1=input1/10;

}

while(input2!=0)

{

n[p]=input2%10;

p++;

input2=input2/10;

}

while(input3!=0)

{

n[p]=input3%10;

p++;

input3=input3/10;

}

while(input4!=0)

{

n[p]=input4%10;

p++;

input4=input4/10;

}

for(i=0;i<20;i++)

f[i]=1;

//freq count and remove duplicate

for(i=0;i<p-1;i++)

{

for(j=i+1;j<p;j++)

{

if((n[i]==n[j])&&(n[i]!=-1))

{

f[i]=f[i]+1;

n[j]=-1;

}

}

}

//delete all -1

for(i=0;i<p;i++)

{

if(n[i]==-1)

{

for(j=i;j<p;j++)

n[j]=n[j+1];

}

}

int max=f[0];

for(i=0;i<p;i++)

{

if(f[i]>=max)

{

s[k]=n[i];

k++;

}

}

int m1=s[0];

for(i=0;i<k;i++)

{

if(s[i]>m1)

m1=s[i];

}

return m1;

}

}

----------------------------------------------------------------------