Product and sum of number is same as original number

for(int i=10;i<=99;i++){

int k=i%10;

int m=i/10;

if(i==(k\*m)+(k+m)){

System.out.println(i);

Even or Odd without Mod

Scanner sc = new Scanner(System.in);

int n= sc.nextInt();

if((n/2)\*2==n){

System.out.println("Even Number");

}else{

System.out.println("Odd number");

       }

2nd method

Scanner sc = new Scanner(System.in);

int n= sc.nextInt();

if((n&1)==0){

System.out.println("Even Number");

}else{

System.out.println("Odd number");

       }

Swap of number using bitwise

Scanner sc = new Scanner(System.in);

int n= sc.nextInt();

int m = sc.nextInt();

n=n^m;

m=n^m;

n=n^m;

System.out.println(n);

System.out.println(m);

Amstrong number

public static void main(String[] args) {

int n = 153;

int count = 0;

int t = n;

while(t!=0){

t = t/10;

count ++;

}

int s =0;

t=n;

while(t!=0){

int d=t%10;

s+=Math.pow(d,count);

t=t/10;

}

if(s==n){

System.out.println("Amstrong number");

}else{

System.out.println("It is not a Amstrong number");

}

Kapraker Value

int n = 297;

int square = n \* n;

int t = square;

int digits = 0;

while (t > 0) {

digits++;

t /= 10;

}

int count = 1;

for (int i = 1; i < digits; i++) {

count = count \* 10;

int l = square / count;

int r = square % count;

if (r > 0 && (l + r == n)) {

System.out.println("It is a Kaprekar number");

}

}

Rotate array by one

public static void main(String[] args) {

int arr[]={1,2,3,4,5};

int n= arr.length;

int last=arr[n-1];

for(int i=n-1;i>0;i--){

arr[i]=arr[i-1];

}

arr[0]=last;

for(int i=0;i<n;i++){

System.out.println(arr[i] +" ");

}

}

}

rotate array by d  
int arr[] = {1, 2, 3, 4, 5};

int n = arr.length;

int last = arr[n - 1];

int d = 2;

int[] t= new int[5];

for (int i = 0; i<d; i++) {

t[i] = arr[n - d + i];

}

for (int i = n-1; i>=d; i--) {

arr[i] = arr[i-d];

}

for (int i = 0; i < d; i++) {

arr[i]=t[i];

}

for (int i = 0; i < n; i++) {

System.out.println(arr[i]);

}

Sending all zeros to last without sorting

int arr[]={1,0,2,0,3,5};

int[] t= new int[6];

int n=arr.length;

int d=6;

int count =0;

for (int i = 0; i<d; i++) {

if(arr[i]!=0){

arr[count++]=arr[i];

}

}

while(count<n){

arr[count++] =0;

}

for(int i=0;i<n;i++){

System.out.println(arr[i]);

}

Sum of Max and Min value in an array

int arr[]={20,30,40,50,100};

int[] t= new int[6];

int n=arr.length;

int max=arr[0];

for (int i = 0; i<n-1; i++) {

if(arr[i+1]>max){

max=arr[i+1];

}

}

int min=arr[0];

for (int i = 0; i<n-1; i++) {

if(arr[i]<min){

min=arr[i+1];

}

}

System.out.println(min+max);

Sliding Window

int arr[]={20,30,40,50,100};

int n=arr.length;

int sum =0;

int target=190;

int k=3;

if(n<k){

System.out.println("The n is less than k"+k);

return;

}

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

sum += arr[i];

}

if(sum==target){

System.out.println("target found");

}

for (int i = k; i<n; i++) {

sum+=arr[i]-arr[i-k];

if(sum==target){

System.out.println("target found ");

}

}

Zig-Zag Pattern

int n=3;

int col=13;

for(int i=1;i<n;i++){

for(int j=1;j<col;j++){

if((i==1&&j%4==3)||(i==2&&j%2==0)||(i==3&&j%4==1)){

System.out.print("\* ");

}else{

System.out.print(" ");

}

}

System.out.println();

}

Fibo using direct recursion

public int fibo(int n){

if(n<=1){

return n;

}

return fibo(n-1)+fibo(n-2);

}

public static void main(String[] args) {

int num=11;

Main t = new Main();

for(int i=0;i<num;i++){

System.out.print(t.fibo(i)+ " ");

}

}

Using indirect recurrison

{

int result=0;

int oddnum(int odd){

if(odd%2!=0){

return result=odd+1;

}else{

return evennum(odd);

}

}

int evennum(int even){

if(even%2==0){

return result=even-1;

}else{

return oddnum(even);

}

}

public static void main(String[] args) {

int num=10;

Main t = new Main();

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

System.out.print(t.oddnum(i)+ " ");

}

}

series of odd+1 and even-1;