A Very Interesting Pattern

class Solution {

public static void Met(int n) {

/\* write your solution here \*/

int a[]={2,2,3,5,7,11,13,17,19,23};

String s=String.valueOf(n);

for(int i=0;i<s.length();i++){

int k=Character.getNumericValue(s.charAt(i));

for(int j=0;j<a[k];j++){

System.out.print("x");

}

System.out.print(k+"\n");

}

}}

An Integer Triangle

class Solution {

public static void Met(int N) {

/\* write your solution here \*/

int c=0;

for(int k=1;k<=N;k++){

c+=k;

}

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

int cu=0;

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

if(j!=0)

cu=cu+(N-j);

if(c-i-cu>=10)

System.out.print(c-i-cu);

else

System.out.print("0"+(c-i-cu));

if(j!=i)

System.out.print(".");

}

System.out.println();

}

}

}

Dr CooCoo's Weird Demands

class Solution {

public static int [] Met(int n) {

/\* write your solution here \*/

int c[]=new int[n\*n];

int j=0;

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

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

c[j]=0;

j++;

}

int l=i;

while(l>0){

c[j]=l;l--;j++;

}

}

return c;

}

}

Dr CooCoo's Weird Demands V2

class Solution {

public static int [] Met(int n) {

/\* write your solution here \*/

int c[]=new int[n\*(n+1)/2];

int j=0;

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

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

c[j]=k;j++;

}

}

return c;

}

}

Spiralling

class Solution {

public static void Met(int[][] a) {

/\* write your solution here \*/

int m=a.length,n=a[0].length,i,k=0,l=0;

while(k<m && l<n){

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

{

System.out.print(a[k][i]+" ");

}

k++;

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

{

System.out.print(a[i][n-1]+" ");

}

n--;

if ( k < m)

{

for (i = n-1; i >= l; --i)

{

System.out.print(a[m-1][i]+" ");

}

m--;

}

if (l < n)

{

for (i = m-1; i >= k; --i)

{

System.out.print(a[i][l]+" ");

}

l++;

}

}

System.out.println();

}}

Mrs Qiqi's Pairs

public static void Met(int a[],int key){

int min=key,k=0,l=1;

int i=0,j=a.length-1;

while(i<j){

if(a[i]+a[j]==key){

k=i;l=j;

i=a.length;

break;

}

else if(key>a[i]+a[j]){

if(min>=key-(a[i]+a[j])){

min=key-(a[i]+a[j]);

k=i;l=j;

}i++;

}

else{

if(min>=(a[i]+a[j])-key){

min=(a[i]+a[j])-key;

k=i;l=j;

}

j--;

}

}

System.out.println(k+" "+l);

}

Mrs Qiqi's i And A[i]

public static void Met(int a[]){

int c=0;

for(int i=0;i<a.length;i++){

if(a[i]==i){

c=i;

break;}

}

if(a[c]==c)

System.out.println(c);

else

System.out.println("No");

}

Subzero

public static void Met(int a[]){

int d=0;

for(int i=0;i<a.length-2&&d==0;i++){

for(int j=i+1;j<a.length-1&&d==0;j++){

for(int k=j+1;k<a.length&&d==0;k++){

if(a[i]+a[j]+a[k]==0&&d==0){

System.out.println(i+" "+j+" "+k);

d=1;

break;

}

}

}

}

if(d==0)

System.out.println("No");

}

Greatest Product In An Array

public static void Met(int [] a){

int great=-1;

for(int i=0;i<a.length-1;i++){

for(int j=i+1;j<a.length;j++){

for(int k=0;k<a.length;k++){

if(a[k]==a[i]\*a[j]){

if(great<a[i]\*a[j]){

great=a[i]\*a[j];

}

}

}

}

}

if(great!=-1)

System.out.println(great);

else

System.out.println(-1);

}

Mr Jack In The Desert

public static int Met(int d){

int s=0,step=0;

int k=Math.abs(d);

while(s<k||(s-k)%2!=0){

step++;

s+=step;

}

return step;

}

Mrs Qiqi's Union

class Solution {

public static void Met(int [] a, int [] b, int key) {

/\* write your solution here \*/

int c[]=new int[a.length+b.length];

int k=0,l=0;

for(int i=0;i<c.length;i++){

if(k<a.length&&l<b.length){

if(a[k]>=b[l]){

c[i]=b[l];

l++;

}

else if(a[k]<b[l]){

c[i]=a[k];

k++;

}

}

else if(k==a.length){

c[i]=b[l];l++;

}else{

c[i]=a[k];k++;

}

}

int temp[]=new int[c.length];

int j=0;

for(int i=0;i<c.length-1;i++){

if(c[i]!=c[i+1]){

temp[j]=c[i];j++;

}

}

System.out.println(temp[key-1]);

}

}

Mrs Qiqi's Union 2

class Solution {

public static void Met(int [] A, int [] B, int [] C, int key) {

/\* write your solution here \*/

int d[]=new int[A.length+B.length];

int k=0,l=0;

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

if(k<A.length&&l<B.length){

if(A[k]<B[l]){

d[i]=A[k];k++;

}

else{

d[i]=B[l];l++;

}

}

else if(k==A.length){

d[i]=B[l];l++;

}

else{

d[i]=A[k];k++;

}

}

int e[]=new int[d.length+C.length];

k=0;l=0;

for(int i=0;i<e.length;i++){

if(k<d.length&&l<C.length){

if(d[k]<C[l]){

e[i]=d[k];k++;

}

else{

e[i]=C[l];l++;

}

}

else if(k==d.length){

e[i]=C[l];l++;

}

else{

e[i]=d[k];k++;

}

}

int temp[]=new int[e.length];

int j=0;

for(int i=0;i<e.length-1;i++){

if(e[i]!=e[i+1]){

temp[j]=e[i];j++;

}

}

if(temp[j]!=e[e.length-1]){

temp[j]=e[e.length-1];j++;

}

System.out.println(temp[key-1]);

}

}