1. 2 strings are anagram

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

isAnagram("seek","peek");

}

static void isAnagram(String S1, String S2)

{

String S11= S1.replaceAll("\\s","");

String S12= S1.replaceAll("\\s","");

boolean status = true;

if(S11.length()!=S12.length())

{

status=false;

}

else

{

char[] ArrayS1 = S11.toLowerCase().toCharArray();

char[] ArrayS2 = S12.toLowerCase().toCharArray();

Arrays.sort(ArrayS1);

Arrays.sort(ArrayS2);

System.out.println("the ArrayS1= "+ Arrays.toString(ArrayS1));

System.out.println("the ArrayS2= "+ Arrays.toString(ArrayS2));

status = Arrays.equals(ArrayS1,ArrayS2);

System.out.println("the status= "+ status);

}

if (status){

System.out.println("the strings "+S1+" and "+S2+"are anagram" );

}

else{

System.out.println("the strings "+S1+" and "+S2+" are not anagram" );

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Check string is panagram or not

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

isPanagram("The quick brown fox jumps over the lazy dog");

}

static void isPanagram(String S1)

{

//Array to store counts of 26 alphbets

int count[] = new int[26];

boolean flag = true; //Assuming sentence is pangram

char ch;

//Loop through each character of a string

for(int i=0; i<S1.length(); i++){

ch = S1.charAt(i);

if(ch==' ')

continue;

//checking if tha character is among 26 alphabets

//If yes then increment the count at their relative position

//'A'-'A' = 0

//'B'-'A' = 1

//'b'-'a' = 1

//'c'-'a' = 2

if(ch>='A' && ch<='Z'){

count[ch-'A']++;

} else if(ch>='a' && ch<='z'){

count[ch-'a']++;

}

}

//checking if count array has any zeros

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

if(count[i] == 0){

flag = false; //Not Pangram so break

break;

}

}

if(flag)

System.out.println("Pangram");

else

System.out.println("Not a Pangram String");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Remove the duplicate words from the given Sentence

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

removeDupwords("The quick brown fox jumps over the lazy dog dog");

}

static void removeDupwords(String S1)

{

//will split the string by space

String[] strWords = S1.split("\\s+");

//convert String array to LinkedHashSet to remove duplicates

LinkedHashSet<String> lhSetWords

= new LinkedHashSet<String>( Arrays.asList(strWords) );

//join the words again by space

StringBuilder sbTemp = new StringBuilder();

int index = 0;

for(String s : lhSetWords){

if(index > 0)

sbTemp.append(" ");

sbTemp.append(s);

index++;

}

S1 = sbTemp.toString();

System.out.println("String after removing duplicate words: ");

System.out.println(S1);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

First non repeating character

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

String S1= "gibblegabbler";

System.out.println("The given string is: " + S1);

nonRepeatingCharacter(S1);

}

static void nonRepeatingCharacter(String S1)

{

for(int i = 0; i < S1.length() ; i++)

{

boolean unique= true;

for(int j = 0; j < S1.length() ; j++)

{

if(i!=j && S1.charAt(i)==S1.charAt(j))

{

unique= false;

break;

}

}

if(unique)

System.out.println("first non duplicate character=="+S1.charAt(i));

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Find a program to find fibonacci series

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*perfect number \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

isPerfect();

}

private static void isPerfect() {

int num, sum = 0;

int i;

for (num = 1; num < 100; num++) {

for (int j = 1; j <= num - 1; j++) { // change made here

if (num % j == 0) {

sum = sum + j;

}

}

if (sum == num) {

System.out.println(sum);

}

sum = 0; // change made here

}

}

}

Strong number:

class IsAStrongNumberOrNot  
{  
    public static void main(String s[])  
    {  
        int input = 145;  
          
        int temp, mod, sum = 0;  
      
        temp = input;  
      
        while(input > 0)  
        {  
            mod = input % 10;  
      
            sum=sum + fact (mod);  
      
            input  = input /10;  
                      
        }  
        if(temp == sum)  
        {  
            System.out.println(temp +" is a strong number");  
        }  
        else  
        {  
            System.out.println(temp +" is not a strong number");  
        }  
    }  
    public static int fact(int num)  
    {  
        int fact=1;  
      
        for(int i = 1; i <= num; i++)  
        {  
            fact \*= i;  
        }  
        return fact;  
    }  
          
}

Complex

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*String as input and maximum occurance of string in format\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Find steps for child:

// Program to find n-th stair

// using step size 1 or 2 or 3.

import java.util.\*;

import java.lang.\*;

public class GfG {

    // Returns count of ways to reach

    // n-th stair using 1 or 2 or 3 steps.

    public static int findStep(int n)

    {

        if (n == 1 || n == 0)

            return 1;

        else if (n == 2)

            return 2;

        else

            return findStep(n - 3) + findStep(n - 2) + findStep(n - 1);

    }

    // Driver function

    public static void main(String argc[])

    {

        int n = 4;

        System.out.println(findStep(n));

    }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Pascal triangle:

filter\_none

edit

play\_arrow

brightness\_4

|  |
| --- |
| // Java code for Pascal's Triangle  import java.io.\*;    class GFG {        // Function to print first      // n lines of Pascal's Triangle      static void printPascal(int n)      {        // Iterate through every line      // and print entries in it      for (int line = 0; line < n; line++)      {          // Every line has number of          // integers equal to line number          for (int i = 0; i <= line; i++)          System.out.print(binomialCoeff                          (line, i)+" ");            System.out.println();      }      }        // Link for details of this function      // <https://www.geeksforgeeks.org/space-and-time-efficient-binomial-coefficient/>      static int binomialCoeff(int n, int k)      {          int res = 1;            if (k > n - k)          k = n - k;            for (int i = 0; i < k; ++i)          {              res \*= (n - i);              res /= (i + 1);          }          return res;      }        // Driver code      public static void main(String args[])      {      int n = 7;      printPascal(n);      }  } |

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Armstrong number

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

int n =153;

isAmstrongNumber(153);

}

private static void isAmstrongNumber(int n)

{

int c=0,a,temp;

temp=n;

while(n>0)

{

a=n%10;

n=n/10;

c=c+a\*a\*a;

System.out.println(" c==="+c);

}

if(c==temp)

System.out.println("the number is Amstrong==="+temp);

else

System.out.println("the number is not Amstrong==="+temp);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Prime factors

import java.util.Scanner;

public class PrimeFactors {

   public static void main(String args[]){

      int number;

      Scanner sc = new Scanner(System.in);

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

      number = sc.nextInt();

      for(int i = 2; i< number; i++) {

         while(number%i == 0) {

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

            number = number/i;

         }

      }

      if(number >2) {

         System.out.println(number);

      }

   }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*max of character \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

String str = "aaaabbddddddddddddddddddddaaccde";

List<String> parts = Arrays.asList(str.split("(?<=(.))(?!\\1)"));

String max = Collections.max(parts, Comparator.comparing(s -> s.length()));

System.out.println("largest substring: " + max);

int index = str.indexOf(max);

System.out.println("index of largest substring: " + index);

largest substring: dddddddddddddddddddd

index of largest substring: 6

median of 2 sorted arrays.

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

int[] A= {1,2} ;

int[] B= {3,4} ;

double dbl=findMedianSortedArrays(A,B);

System.out.println("dbl==="+dbl);

}

static double findMedianSortedArrays(int[] A, int[] B) {

int m = A.length;

int n = B.length;

if (m > n) { // to ensure m<=n

int[] temp = A; A = B; B = temp;

int tmp = m; m = n; n = tmp;

}

int iMin = 0, iMax = m, halfLen = (m + n + 1) / 2;

while (iMin <= iMax) {

int i = (iMin + iMax) / 2;

int j = halfLen - i;

if (i < iMax && B[j-1] > A[i]){

iMin = i + 1; // i is too small

}

else if (i > iMin && A[i-1] > B[j]) {

iMax = i - 1; // i is too big

}

else { // i is perfect

int maxLeft = 0;

if (i == 0) { maxLeft = B[j-1]; }

else if (j == 0) { maxLeft = A[i-1]; }

else { maxLeft = Math.max(A[i-1], B[j-1]); }

if ( (m + n) % 2 == 1 ) { return maxLeft; }

int minRight = 0;

if (i == m) { minRight = B[j]; }

else if (j == n) { minRight = A[i]; }

else { minRight = Math.min(B[j], A[i]); }

return (maxLeft + minRight) / 2.0;

}

}

return 0.0;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Fibonacci example

[**next →← prev**](https://www.javatpoint.com/java-programs)

Fibonacci series in Java

In fibonacci series, *next number is the sum of previous two numbers* for example 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 etc. The first two numbers of fibonacci series are 0 and 1.

There are two ways to write the fibonacci series program in java:

* Fibonacci Series without using recursion
* Fibonacci Series using recursion

Fibonacci Series in Java without using recursion

Let's see the fibonacci series program in java without using recursion.

1. **class** FibonacciExample1{
2. **public** **static** **void** main(String args[])
3. {
4. **int** n1=0,n2=1,n3,i,count=10;
5. System.out.print(n1+" "+n2);//printing 0 and 1
7. **for**(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are already printed
8. {
9. n3=n1+n2;
10. System.out.print(" "+n3);
11. n1=n2;
12. n2=n3;
13. }
15. }}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*complex: count of chars in string\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

String S1="aabccd";

stringManipulation(S1);

}

static void stringManipulation(String str)

{

//ASCII values ranges upto 256

int counter[] = new int[256];

//String length

int len = str.length();

/\* This array holds the occurrence of each char, For example

\* ASCII value of A is 65 so if A is found twice then

\* counter[65] would have the value 2, here 65 is the ASCII value

\* of A

\*/

for (int i = 0; i < len; i++)

counter[str.charAt(i)]++;

// We are creating another array with the size of String

char array[] = new char[str.length()];

for (int i = 0; i < len; i++)

{

array[i] = str.charAt(i);

int flag = 0;

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

/\* If a char is found in String then set the flag

\* so that we can print the occurrence

\*/

if (str.charAt(i) == array[j])

flag++;

}

if (flag == 1)

System.out.println("Occurrence of char " + str.charAt(i)

+ " in the String is:" + counter[str.charAt(i)]);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*shortest array and size^\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

int arr[][]={ {1,2,3,4},{0,2,7},{4,12},{1,2,5,6,2}};

shortArray(arr);

}

static void shortArray(int arr [][])

{

int max =0;

int arrlen[]= new int[arr.length] ;

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

{

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

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

arrlen[i] = arr[i].length;

}

System.out.println("debashree1"+ Arrays.toString(arrlen));

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***MEDIUM**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1)Armstrong number

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

//int n=100;

isAmstrong();

}

public static void isAmstrong()

{

int c=0,temp,a;

int n=153;//It is the number to check armstrong

temp=n;

while(n>0)

{

a=n%10;

n=n/10;

c=c+a\*a\*a;

}

if(temp==c)

System.out.println("This number is armstrong=="+temp);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*numerology\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/.

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

String input = "S. KANAPATHY";

numerology(input);

}

static void numerology(String input)

{

String strNumerology = input.replaceAll("\\s", "");

strNumerology =strNumerology.replaceAll("[^a-zA-Z0-9\_-]","");

System.out.println("strNumerology==="+strNumerology);

int sum=0;

int a=0;

int b=0;

int c=0;

int d=0;

int e=0;

int f=0;

int g=0;

int h=0;

for(int n=0; n < strNumerology.length() ; n++)

{

if(strNumerology.charAt(n)==('A')|| strNumerology.charAt(n)==('J') || strNumerology.charAt(n)==('Q') || strNumerology.charAt(n)==('Y') || strNumerology.charAt(n)==('I'))

{

a=1;

//sum=sum+a;

}

else if(strNumerology.charAt(n)==('B')|| strNumerology.charAt(n)==('K') || strNumerology.charAt(n)==('R'))

{

b=2;

//sum=sum+b;

}

else if(strNumerology.charAt(n)==('C')|| strNumerology.charAt(n)==('G') || strNumerology.charAt(n)==('L') || strNumerology.charAt(n)==('S'))

{

c=3;

//sum=sum+c;

}

else if(strNumerology.charAt(n)==('D')|| strNumerology.charAt(n)==('M') || strNumerology.charAt(n)==('T'))

{

d=4;

//sum=sum+d;

}

else if(strNumerology.charAt(n)==('E')|| strNumerology.charAt(n)==('H') || strNumerology.charAt(n)==('X'))

{

e=5;

//sum=sum+e;

}

else if(strNumerology.charAt(n)==('U')|| strNumerology.charAt(n)==('V') || strNumerology.charAt(n)==('W'))

{

f=6;

//sum=sum+f;

}

else if(strNumerology.charAt(n)==('O')|| strNumerology.charAt(n)==('Z') )

{

g=7;

//sum=sum+g;

}

else if(strNumerology.charAt(n)==('B')|| strNumerology.charAt(n)==('F') || strNumerology.charAt(n)==('P'))

{

h=8;

//sum=sum+h;

}

sum= a+b+c+d+e+f+g+h;

}

System.out.println("sum of the input string"+sum);

int z=0;

int num=0;

while(sum>10)

{

z= sum%10;

num = z+num;

}

System.out.println("numerology==="+num);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*prime factors\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Program to print all prime factors

import java.io.\*;

import java.lang.Math;

class GFG {

    // A function to print all prime factors

    // of a given number n

    public static void primeFactors(int n)

    {

        // Print the number of 2s that divide n

        while (n % 2 == 0) {

            System.out.print(2 + " ");

            n /= 2;

        }

        // n must be odd at this point.  So we can

        // skip one element (Note i = i +2)

        for (int i = 3; i <= Math.sqrt(n); i += 2) {

            // While i divides n, print i and divide n

            while (n % i == 0) {

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

                n /= i;

            }

        }

        // This condition is to handle the case whien

        // n is a prime number greater than 2

        if (n > 2)

            System.out.print(n);

    }

    public static void main(String[] args)

    {

        int n = 315;

        primeFactors(n);

    }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*median of 2 sorted arrays\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args) {

int arr1[] = {2,3,6,7,9};

int arr2[] = {1,4,8,10};

double k= findMedianSortedArrays(arr1,arr2);

System.out.println("median==="+k);

}

static double findMedianSortedArrays(int[] nums1, int[] nums2) {

int total = nums1.length+nums2.length;

if(total%2==0){

return (getKth(nums1, 0, nums1.length-1, nums2, 0, nums2.length-1, total/2)

+ getKth(nums1, 0, nums1.length-1, nums2, 0, nums2.length-1, total/2-1))/2.0;

}else{

return getKth(nums1,0, nums1.length-1, nums2, 0, nums2.length-1, total/2);

}

}

//k is the index starting from 0

static int getKth(int[] nums1, int i1, int j1, int[] nums2, int i2, int j2, int k){

if(j1<i1){

return nums2[i2+k];

}

if(j2<i2){

return nums1[i1+k];

}

if(k==0){

return Math.min(nums1[i1], nums2[i2]);

}

int len1 = j1 - i1 + 1;

int len2 = j2 - i2 + 1;

int m1 = k\*len1/(len1+len2);

int m2 = k - m1 - 1;

m1 += i1;

m2 += i2;

if(nums1[m1]<nums2[m2]){

k = k-(m1-i1+1);

j2 = m2;

i1 = m1+1;

}else{

k = k-(m2-i2+1);

j1 = m1;

i2 = m2+1;

}

return getKth(nums1, i1, j1, nums2, i2, j2, k);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;  
import java.util.stream.Collectors;  
  
class Student{  
    private String name;  
    private Float averageMark = 0.0f;  
    private int numberOfSubjects = 0;  
  
    public Student(String name){  
        this.name = name;  
    }  
  
    public void addMark(Float mark ){  
        averageMark += mark;  
        numberOfSubjects += 1;  
        averageMark = averageMark/numberOfSubjects;  
    }  
  
    public Float getAverageMark(){  
        return averageMark;  
    }  
  
    public String getName(){  
        return name;  
    }  
  
    public Student(Float mark, String n){  
        name = n;  
        averageMark = mark;  
    }  
  
}  
  
public class MaxScore {  
    public static String s[][] = {{"jerry","65"},  
            {"bob","91"}, {"jerry","23"}, {"Eric","83"},{"Eric","99"}};  
  
    public static void main(String args[]){  
        Map studentMap = new HashMap();  
        for(int i=0; i studentList = studentMap.values().stream().collect(Collectors.toList());  
        Collections.sort(studentList, Comparator.comparing(Student::getAverageMark).reversed());  
  
        System.out.println(studentList.get(0).getName());  
  
    }  
  
}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*mimum sum\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Util

{

// Count ways to calculate a target from elements of specified array

public static int countWays(int[] arr, int n, int target)

{

// base case: if target is found

if (target == 0)

return 1;

// base case: No elements are left

if (n < 0)

return 0;

// 1. Ignore current element

int exclude = countWays(arr, n - 1, target);

// 2. Consider current element

// 2.1. Subtract current element from the target

// 2.2. Add current element to the target

int include = countWays(arr, n - 1, target - arr[n]) +

countWays(arr, n - 1, target + arr[n]);

// Return total count

return exclude + include;

}

public static void main(String[] args)

{

// input array and target number

int[] arr = { 5, 3, -6, 2 };

int target = 6;

System.out.println(countWays(arr, arr.length - 1, target) + " ways");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Find out unique substring of given length of string.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args)

{

String s = "aab";

int n = 2;

// Function calling

result(s, n);

}

// Function to print the

// unique sub-String of length n

static void result(String s1,int n)

{

// set to store the Strings

HashSet<String> st = new HashSet<String>();

for (int i = 0; i < s1.length(); i++)

{

String ans = "";

for (int j = i; j < s1.length(); j++)

{

ans += s1.charAt(j);

// if the size of the String

// is equal to 1 then insert

if (ans.length()== n)

{

// inserting unique

// sub-String of length L

st.add(ans);

break;

}

}

}

// Printing the set of Strings

for (String it : st)

System.out.print(it + " ");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Fliping coins complex\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*code to print pascal triangle for row=5 and co-0rdinate\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import java.util.\*;

/\*

\* To execute Java, please define "static void main" on a class

\* named Solution.

\*

\* If you need more classes, simply define them inline.

\*/

class Solution {

public static void main(String[] args)

{

int rows = 5;

printPascal(rows);

}

static void printPascal(int n)

{

for(int line = 1; line <= n; line++)

{

int C=1;

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

{

System.out.print(C+" ");

C = C \* (line - i) / i;

}

System.out.println();

}

long nCordinate = GetPasVal(3,2);

System.out.println("co-ordinate===" + nCordinate);

}

static long GetPasVal(int height, int width)

{

long[][] triangle = new long[height][];

for (int i = 0; i < height; i++)

{

triangle[i] = new long[i + 1];

triangle[i][0] = 1;

triangle[i][i] = 1;

if (i >= 2)

{

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

{

triangle[i][j] = triangle[i - 1][j - 1] + triangle[i - 1][j];

}

}

}

return triangle[height - 1][width - 1];

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*8check given number is strong number or not (medium)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class SubArrayWithGivenSum {

public void findSubArray(int[] arrA, int Sum) {

int currSum = 0;

int start = 0;

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

while (currSum > Sum) {

// System.out.println(currSum + " reducing");

currSum = currSum - arrA[start++];

// System.out.println(currSum + " reduced");

}

if (currSum == Sum) {

System.out.println(Sum + " is found between indexes " + start

+ " and " + (i - 1));

// print the exact elements

System.out.print("And Elements are : ");

for (int j = start; j <= i - 1; j++) {

System.out.print(" " + arrA[j]);

}

return;

}

if (i < arrA.length)

currSum = currSum + arrA[i];

// System.out.println(currSum);

}

// if we have reached that means that we have not found the subarray

// with summation equals to Sum

System.out.println("No subarray is found with sum equals to " + Sum);

}

public static void main(String[] args) throws java.lang.Exception {

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

int Sum = 6;

SubArrayWithGivenSum i = new SubArrayWithGivenSum();

i.findSubArray(arrA, Sum);

}

}