**Complex Program:**

String Manipulation Program: string as input like 'aabccd' and print the maximum occurrence of a string in the format a=2,b=1,c=2,d=1

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)]);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* the shortest array and its length from the the array group.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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}};

shortArray(arr);

}

static void shortArray(int arr[][])

{

int max =0; int arrlen[]= new int[arr.length] ;

int prevLen =0;

int currLen =0;

int minlen =0;

int flag =0;

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

{

currLen =arr [i].length;

// System.out.println("prevLen"+ prevLen);

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

if(currLen > prevLen)

{

minlen = prevLen;

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

}

if(arr[i].length == minlen)

flag=1;

prevLen=currLen;

}

if(flag==1)

{

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

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* pascal triangle and co-ordinates\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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];

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* string array, each element containing a line of apache log\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import java.util.\*;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

/\*

\* 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) {

final String log = "123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] \"GET /pics/wpaper.gif HTTP/1.0\" 200 6248 \"http:// www.jafsoft.com/asctortf/\" \"Mozilla/4.05 (Macintosh; I; PPC)\"\n"

+ "123.123.123.123 - - [26/Apr/2000:00:23:47 -0400] \"GET /asctortf/ HTTP/1.0\" 200 8130 \"http:// search.netscape.com/Computers/Data\_Formats/Document/Text/RTF\" \"Mozilla/4.05 (Macintosh; I; PPC)\"\n"

+ "123.123.123.124 - - [26/Apr/2000:00:23:48 -0400] \"GET /pics/5star2000.gif HTTP/1.0\" 200 4005 \"http:// www.jafsoft.com/asctortf/\" \"Mozilla/4.05 (Macintosh; I; PPC)\"\n"

+ "123.123.123.123 - - [26/Apr/2000:00:23:50 -0400] \"GET /pics/5star.gif HTTP/1.0\" 404 1031 \"http:// www.jafsoft.com/asctortf/\" \"Mozilla/4.05 (Macintosh; I; PPC)\"\n"

+ "123.123.123.126 - - [26/Apr/2000:00:23:51 -0400] \"GET /pics/a2hlogo.jpg HTTP/1.0\" 200 4282 \"http:// www.jafsoft.com/asctortf/\" \"Mozilla/4.05 (Macintosh; I; PPC)\"\n"

+ "123.123.123.123 - - [26/Apr/2000:00:23:51 -0400] \"GET /cgi-bin/newcount?jafsof3&width=4&font=digital&noshow HTTP/1.0\" 200 36 \"http:// www.jafsoft.com/asctortf/\" \"Mozilla/4.05 (Macintosh; I; PPC)\"\n";

findMaxIpCount(log);

}

static void findMaxIpCount(String record)

{

// Creating a regular expression for the records

final String regex = "^(\\S+) (\\S+) (\\S+) " +

"\\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(\\S+)" +

" (\\S+)\\s\*(\\S+)?\\s\*\" (\\d{3}) (\\S+)";

final Pattern pattern = Pattern.compile(regex, Pattern.MULTILINE);

final Matcher matcher = pattern.matcher(record);

// Creating a Hashmap containing string as

// the key and integer as the value.

HashMap<String, Integer> countIP = new HashMap<String, Integer>();

while (matcher.find()) {

String IP = matcher.group(1);

String Response = matcher.group(8);

int response = Integer.parseInt(Response);

// Inserting the IP addresses in the

// HashMap and maintaining the frequency

// for each HTTP 200 code.

if (response == 200) {

if (countIP.containsKey(IP)) {

countIP.put(IP, countIP.get(IP) + 1);

}

else {

countIP.put(IP, 1);

}

}

}

// Printing the hashmap

for (Map.Entry entry : countIP.entrySet()) {

System.out.println(entry.getKey() + " " + entry.getValue());

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*nth stairs without\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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 s = 4;

System.out.println("Number of ways = "+ countWays(s));

}

// A simple recursive program to find n'th fibonacci number

static int fib(int n)

{

if (n <= 1)

return n;

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

}

// Returns number of ways to reach s'th stair

static int countWays(int s)

{

return fib(s + 1);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*skip steps \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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 = 4;

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

}

// 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);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*flipping coins \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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[] coins = { 10, 2, 5 };

int[] count = { 5, 10, 50 };

int sum = 50;

int n = countWays(coins, count, sum);

System.out.println(n);

}

/\*Coin change problem with finite number of coins available denominations\*/

static int countWays(int[] coins, int[] count, int sum)

{

int n = coins.length;

int[][] dp = new int[sum + 1][n + 1];

int ret = 0;

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

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

for (int k = 1; k <= count[j - 1]; k++) {

if (i > coins[j - 1] \* k)

dp[i][j] += dp[i - coins[j - 1] \* k][j - 1];

if (i == coins[j - 1] \* k)

dp[i][j] += 1;

}

}

}

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

ret += dp[sum][i];

}

return ret;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*weight of given string\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **int** findWeigthOfString(String input1,**int** input2){

**int** sum=0;

String s=input1.toUpperCase;

**int** len=input1.length();

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

**char** letter=s.charAt(i);

**if**(Character.isAlphabetic(letter)){

**if**((input2==0)&&(letter=='A'||letter=='E'||letter=='I'||letter=='O'||letter=='U'))

sum+=0

**else**

sum+=letter-64;

}

}

**return** sum;

}

ASCII values

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 Str1 = "Apple";

weightString(Str1);

}

static void weightString(String s)

{

int sum\_char = 0;

// loop to sum the ascii value of chars

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

System.out.println("heee==="+(int)s.charAt(i));

sum\_char += (int)s.charAt(i);

}

System.out.println("weight of string =="+sum\_char );

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*weight 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 Str1 = "Apple";

weightString(Str1);

}

static void weightString(String str)

{

String s= str.toLowerCase();

int sum\_char = 0;

int weight =0;

// loop to sum the ascii value of chars

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

if(s.charAt(i)=='a')

weight = 1;

if(s.charAt(i)=='p')

weight = 16;

if(s.charAt(i)=='l')

weight = 12;

if(s.charAt(i)=='e')

weight = 5;

sum\_char += weight;

System.out.println("weight of stringsdsd =="+weight );

}

System.out.println("weight of string =="+sum\_char );

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*all possible function to get given 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 = Integer.parseInt("3");

partition(N);

}

public static void partition(int n) {

partition(n, n, "");

}

public static void partition(int n, int max, String prefix) {

if (n == 0) {

System.out.println(prefix);

return;

}

for (int i = Math.min(max, n); i >= 1; i--) {

partition(n-i, i, prefix + " " + i);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*power off \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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 test\_no = 25;

if(isPowerOfFour(test\_no) == 1)

System.out.println(test\_no +

" is a power of 5");

else

System.out.println(test\_no +

"is not a power of 5");

}

// Function to check if

// x is power of 4

static int isPowerOfFour(int n)

{

if(n == 0)

return 0;

while(n != 1)

{

if(n % 5 != 0)

return 0;

n = n / 5;

}

return 1;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*simple\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*duplicate words\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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 = "Good day day bye bye";

removeDuplicateWords(S1);

}

static void removeDuplicateWords(String duplicateWords)

{

String str = duplicateWords;

System.out.println("Original String: ");

System.out.println(str);

/\*

\* Since the words are separated by space,

\* we will split the string by one or more space

\*/

String[] strWords = str.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++;

}

str = sbTemp.toString();

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

System.out.println(str);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Reverse string preserving order\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. **public** **class** ReverseStringPreserveSpace {
2. **static** **void** reverseString(String input) {
4. **char**[] inputArray = input.toCharArray();
5. **char**[] result = **new** **char**[inputArray.length];
7. **for** (**int** i = 0; i < inputArray.length; i++) {
8. **if** (inputArray[i] == ' ') {
9. result[i] = ' ';
10. }
11. }
13. **int** j = result.length - 1;
15. **for** (**int** i = 0; i < inputArray.length; i++) {
16. **if** (inputArray[i] != ' ') {
17. **if** (result[j] == ' ') {
18. j--;
19. }
20. result[j] = inputArray[i];
21. j--;
22. }
23. }
24. System.out.println(input + " --> " + String.valueOf(result));
25. }
27. **public** **static** **void** main(String[] args) {
28. reverseString("India Is my country");
29. }
30. }

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Fiboonacci series\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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. }}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* a program to find first non-repeating character from a input string\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

public class Main {

public static void main(String[] args) {

String str1 = "gibblegabbler";

System.out.println("The given string is: " + str1);

for (int i = 0; i < str1.length(); i++) {

boolean unique = true;

for (int j = 0; j < str1.length(); j++) {

if (i != j && str1.charAt(i) == str1.charAt(j)) {

unique = false;

break;

}

}

if (unique) {

System.out.println("The first non repeated character in String is: " + str1.charAt(i));

break;

}

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* second smallest number from a array of numbers recursively\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. **public** **class** SecondSmallestInArrayExample{
2. **public** **static** **int** getSecondSmallest(**int**[] a, **int** total){
3. **int** temp;
4. **for** (**int** i = 0; i < total; i++)
5. {
6. **for** (**int** j = i + 1; j < total; j++)
7. {
8. **if** (a[i] > a[j])
9. {
10. temp = a[i];
11. a[i] = a[j];
12. a[j] = temp;
13. }
14. }
15. }
16. **return** a[1];//2nd element because index starts from 0
17. }
18. **public** **static** **void** main(String args[]){
19. **int** a[]={1,2,5,6,3,2};
20. **int** b[]={44,66,99,77,33,22,55};
21. System.out.println("Second smallest: "+getSecondSmallest(a,6));
22. System.out.println("Second smallest: "+getSecondSmallest(b,7));
23. }}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* print all String which starts as "S" from the given String array\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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 = {"Apple","Sample", "Search","Cat"};

String search = "S";

searchString(S1,search);

}

static void searchString(String [] S1,String search)

{

String S2[]= new String [4];

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

{

if(S1[i].startsWith(search.toUpperCase()))

{

//MySortStrings.add(MyStrings.get(i));

System.out.println("string =="+S1[i]);

S2[i] = S1[i];

}

}

System.out.println("final array ==" + (Arrays.toString(S2)));

}

}