[**Viết chương trình tính số Fibonacci thứ n (với n không quá 92)** 4](#_Toc61473432)

[**2 Hãy đưa ra ước số nguyên tố lớn nhất của N** 4](#_Toc61473433)

[**3 Ví dụ S =”ACCBA10D2EW30” ta nhận được kết quả: “AABCCDEW6”.** 5](#_Toc61473434)

[**4 Một số được coi là đẹp nếu đó là số thuận nghịch và chỉ toàn các chữ số chẵn Y/N** 6](#_Toc61473435)

[**5 Hãy viết chương trình đưa các xâu họ tên về dạng chuẩn** 7](#_Toc61473436)

[**6 Hãy tạo ra các địa email tương ứng** 8](#_Toc61473437)

[**7 Chẳng hạn xâu “aabcc” có thể trở thành “bcc” hoặc “aab” sau 1 lần xóa** 10](#_Toc61473438)

[**8 XÂU NHỊ PHÂN KẾ TIẾP** 11](#_Toc61473439)

[**9 Hoán vị kế tiếp. Ví dụ N=5, X[] = {1, 2, 3, 4, 5} thì hoán vị tiếp theo của X[] là {1, 2, 3, 5, 4}** 12](#_Toc61473440)

[**10 Ví dụ với N = 3 ta có kết quả: 321, 312, 231, 213, 132, 123** 13](#_Toc61473441)

[**11 Hãy đưa ra tất cả các tập con của xâu ký tự S theo thứ tự tăng dần của các xâu ký tự** 14](#_Toc61473442)

[**12 Với 1 tổ hợp chập K phần tử của N, hãy cho biết tổ hợp tiếp theo sẽ có bao nhiêu phần tử mới** 16](#_Toc61473443)

[**13 Bạn hãy lựa chọn các công việc một cách tối ưu sao cho số công việc làm được là nhiều nhất. out 4** 18](#_Toc61473444)

[**14 Hãy tính xem cần ít nhất bao nhiêu thời gian để có thể nối xong N sợi dây?** 20](#_Toc61473445)

[**15 DI CHUYỂN TRONG MÊ CUNG 1 chỉ D-R** 21](#_Toc61473446)

[**21  đếm số cách đặt n quân hậu này lên bàn cờ sao cho với 2 quân hậu bất kì, chúng không “ăn” nhau.** 24](#_Toc61473447)

[**22 Nhiệm vụ của bạn là đặt 8 quân hậu lên bàn cờ, sao cho không có 2 quân nào ăn nhau, và số điểm đạt được là lớn nhất.** 26](#_Toc61473448)

[**23  Hãy tìm cách trả ít tờ nhất với số tiền đúng bằng S** 28](#_Toc61473449)

[**24 Ví dụ với N = 5 ta có kết quả là: (5), (4, 1), (3, 2), (3, 1, 1), (2, 2, 1), (2, 1, 1, 1), (1, 1, 1, 1, 1** 29](#_Toc61473450)

[**25 Ví dụ S=”ABC” ta có kết quả {ABC ACB BAC BCA CAB CBA}** 31](#_Toc61473451)

[**26 Ví dụ với n = 9, k = 3, s = 23, {6, 8, 9} là tập hợp duy nhất thỏa mãn** 32](#_Toc61473452)

[**31 Hãy thực hiện thuật toán sắp xếp đổi chỗ trực tiếp trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán** 34](#_Toc61473453)

[**32 Hãy thực hiện thuật toán sắp xếp chọn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.** 35](#_Toc61473454)

[**33 Hãy thực hiện thuật toán sắp xếp chèn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán** 36](#_Toc61473455)

[**34 thuật toán sắp xếp nổi bọt** 37](#_Toc61473456)

[**35 Cho số nguyên dương N và K. Hãy tính NK modulo 109+7.** 38](#_Toc61473457)

[**36 Ví dụ với 2 bước biến đổi, ta có [1] à [1 2 1] à [1 2 1 3 1 2 1].(gấp đôi dãy số)** 39](#_Toc61473458)

[**38 Nhiệm vụ của bạn là hãy xác định số Fibonaci thứ n. Do đáp số có thể rất lớn, in ra kết quả theo modulo 10^9+7** 40](#_Toc61473459)

[**39 Cho một xâu ký tự str bao gồm nhiều từ trong xâu. Hãy đảo ngược từng từ trong xâu?** 41](#_Toc61473460)

[**40 Cho một xâu chỉ gồm các kí tự ‘(‘, ‘)’, ‘[‘, ‘]’, ‘{‘, ‘}’. Một dãy ngoặc đúng** 42](#_Toc61473461)

[**44 Hãy viết chương trình chuyển đổi biểu thức biểu diễn dưới dạng tiền tố về dạng hậu tố \*+AB-CD-> AB+CD-\*** 43](#_Toc61473462)

[**45 chuyển đổi biểu thức biểu diễn dưới dạng hậu tố về dạng tiền tố.** 44](#_Toc61473463)

[**46 Hãy viết chương trình chuyển tính toán giá trị của biểu thức hậu tố 231\*+9–=-4** 45](#_Toc61473464)

[**47 tính toán giá trị của biểu thức tiền tố** 46](#_Toc61473465)

[**49 Cho dãy ký tự S chỉ bao gồm các ký tự I hoặc D. Ký tự I được hiểu là tăng (Increasing) ký tự D được hiểu là giảm (Decreasing). Sử dụng các số từ 1 đến 9, hãy đưa ra số nhỏ nhất được đoán nhận từ S. Chú ý, các số không được phép lặp lại** 48](#_Toc61473466)

[**50 Hãy in ra tất cả các số nhị phân từ 1 đến n** 49](#_Toc61473467)

[**51 có bao nhiêu số BDN không lớn hơn N. Ví dụ N=100 ta có 4 số BDN bao gồm các số: 1, 10, 11, 100** 51](#_Toc61473468)

[**52 Số BDN của N là số P =M´N sao cho P là số BDN. Cho số tự nhiên N (N<1000), hãy tìm số BDN nhỏ nhất của N** 52](#_Toc61473469)

[**53 Hãy tìm số nguyên dương X nhỏ nhất được tạo bởi số 9 và số 0 chia hết cho N. Ví dụ với N = 5 ta sẽ tìm ra X = 90** 53](#_Toc61473470)

[**54 Hãy dịch chuyển S thành T sao cho số lần thực hiện các thao tác (a), (b) là ít nhất** 55](#_Toc61473471)

[**56 Hãy tìm xâu con chung dài nhất của 2 xâu này** 56](#_Toc61473472)

[**61 Cho xâu ký tự S. Nhiệm vụ của bạn là tìm độ dài dãy con lặp lại dài nhất trong S** 57](#_Toc61473473)

[public class **DayConTangDaiNhat** { 58](#_Toc61473474)

[**64 Nhiệm vụ của bạn là tìm tổng lớn nhất của dãy con được sắp theo thứ tự tăng dần của dãy A[]** 59](#_Toc61473475)

[65 Nhiệm vụ của bạn **là sắp xếp lại mảng số với số lượng bước là ít nhất** 60](#_Toc61473476)

[**66 Một con ếch có thể nhảy 1, 2, 3 bước** 61](#_Toc61473477)

[**67 Cho 2 số nguyên n, k. Bạn hãy tính C(n, k) modulo 109+7.** 62](#_Toc61473478)

[**68 Một chiếc cầu thang có N bậc** 63](#_Toc61473479)

[**69 Hãy tìm xâu con đối xứng dài nhất của S** 64](#_Toc61473480)

[**72 Viết ký tự giống nhau .Ví dụ với N = 9, X =1, Y = 2, Z =1 ta có kết quả là 5 bằng cách thực hiện: insert, insert, copying, copying, inser** 65](#_Toc61473481)

[**73 Một bản tin M đã mã hóa bí mật thành các con số theo ánh xạ như sau: ‘A’->1, ‘B’->2, .., ‘Z’->26** 66](#_Toc61473482)

[**74 Nhiệm vụ của bạn là tìm số lượng ít nhất các số nhỏ hơn N mà có tổng bình phương bằng N** 67](#_Toc61473483)

[**ChuyenDanhSachCanhSangDanhSachKe** 68](#_Toc61473484)

[**ChuyenDanhSachKeSangDanhSachCanh** 69](#_Toc61473485)

[public class **DayConCoTongBangS** { 71](#_Toc61473486)

[public class **CaiTui** { 72](#_Toc61473487)

[public class **BieuDienDoThiCoHuong** { 73](#_Toc61473488)

[public class **Bfs** { 74](#_Toc61473489)

[public class **Dfs** { 75](#_Toc61473490)

**Viết chương trình tính số Fibonacci thứ n (với n không quá 92)**

public class No1 {

public static void main(String[] args) {

long f[] = new long[93];

f[1] = 1; f[2] = 1;

for(int i = 3; i< 93; i++) f[i] = f[i-1]+f[i-2];

Scanner sc = new Scanner(System.in);

int total = sc.nextInt();

while(total > 0){

System.out.println(f[sc.nextInt()]);

total--;

}

}

}

**2 Hãy đưa ra ước số nguyên tố lớn nhất của N**

import java.util.Scanner;

public class Bai2 {

public static boolean UocNgto(long num) {

long i = 2;

double mid = Math.sqrt(num);

while (i <= mid && num % i != 0) {

i++;

}

return i > mid ? true : false;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

while (totalRecords-- > 0) {

long num = sc.nextLong();

if (UocNgto(num)) {

System.out.println(num);

} else {

long i = 2;

while (i <= num/2) {

while (num % i != 0) i++;

if (UocNgto(num/i)) {

System.out.println(num / i);

i = num;}

else i++;}}}}}

**3 Ví dụ S =”ACCBA10D2EW30” ta nhận được kết quả: “AABCCDEW6”.**

public class Bai3 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

sc.nextLine();

while (totalRecords-- > 0) {

int total = 0;

ArrayList<Character> letter = new ArrayList<>();

String str = sc.nextLine();

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

if(!Character.isDigit(str.charAt(i))) letter.add(str.charAt(i));

else total+= Character.getNumericValue(str.charAt(i));

}

Collections.sort(letter);

for(Character c : letter){

System.out.print(c);

}

System.out.print(total);

System.out.println();

}

}

}

**4 Một số được coi là đẹp nếu đó là số thuận nghịch và chỉ toàn các chữ số chẵn Y/N**

public class Bai4 {

public static boolean SoDep(String s){

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

if(Character.getNumericValue(s.charAt(i)) % 2 !=0 ) return false;

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

sc.nextLine();

while (totalRecords-- > 0) {

String str = sc.nextLine();

System.out.println(SoDep(str)?"YES":"NO");

} } }

**5 Hãy viết chương trình đưa các xâu họ tên về dạng chuẩn**

public class Bai5 {

public static String ChuanHoa(String s){

String result = ""+s.charAt(0);

result = result.toUpperCase();

s = s.toLowerCase();

return (result + s.substring(1));

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

sc.nextLine();

while (totalRecords-- > 0) {

String str = sc.nextLine();

str = str.trim();

while(str.indexOf(" ") != -1) str = str.replaceAll(" ", " ");

String fullname[] = str.split(" ");

for(String s : fullname){

System.out.print(ChuanHoa(s)+" ");

}

System.out.println();

}

}

}

**6 Hãy tạo ra các địa email tương ứng**

public class Bai6 {

public static ArrayList<String> createMail(ArrayList<String> list){

int no[] = new int[list.size()];

no[1] = 1;

int i = 0;

while (i<list.size()){

int order = 1;

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

if(list.get(i).equals(list.get(j))) order++;

}

no[i] = order;

i++;

}

for(int j = 0; j < list.size(); j++){

if(no[j]>1){

list.set(j, (list.get(j)+no[j]) );

}

}

return list;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

sc.nextLine();

ArrayList<String> list = new ArrayList<>();

while (totalRecords-- > 0) {

String str = sc.nextLine();

str = str.trim();

str = str.toLowerCase();

while(str.indexOf(" ") != -1) str = str.replaceAll(" ", " ");

String fullname[] = str.split(" ");

String result = fullname[fullname.length-1];

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

result += fullname[i].charAt(0);

}

list.add(result);

}

for(String s: createMail(list)){

System.out.println(s+"@ptit.edu.vn");

}

}

}

**7 Chẳng hạn xâu “aabcc” có thể trở thành “bcc” hoặc “aab” sau 1 lần xóa**

public class Bai7 {

public static String RutGonXau(String s) {

if (s.length() == 0) {

return "Empty String";

} else {

int i = 0;

while ((i < s.length() - 1) && (s.charAt(i) != s.charAt(i + 1))) {

i++;

}

if (i == s.length() - 1) {

return s;

}

return RutGonXau(s.substring(0, i) + s.substring(i + 2));

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String str = sc.nextLine();

System.out.println(RutGonXau(str));

}

}

**8 XÂU NHỊ PHÂN KẾ TIẾP**

public class Bai8 {

public static boolean isFinal(int[] a){

for(int i: a) if(i==0) return false;

return true;

}

public static int[] getNext(int[] a){

if(isFinal(a)){

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

return a;

}

else{

int i = a.length-1;

while (a[i]==1) i--;

a[i]=1;

for(int j = i+1; j<a.length; j++) a[j]=0;

return a;

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

sc.nextLine();

while (totalRecords-- > 0) {

String str = sc.nextLine();

int a[] = new int[str.length()];

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

a[i] = Character.getNumericValue(str.charAt(i));

}

for(int i: getNext(a))System.out.print(i);

System.out.println();

}}}

**9 Hoán vị kế tiếp. Ví dụ N=5, X[] = {1, 2, 3, 4, 5} thì hoán vị tiếp theo của X[] là {1, 2, 3, 5, 4}**

public class Bai9 {

public static boolean check(int[] a, int n){

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

if(a[i]!=n-i+1) return false;

}

return true;

}

public static int[] HoanVi(int[] a, int n){

if(check(a,n)){

for(int i = 1; i < n; i++) a[i] = n-i+1;

return a;

}

else{

int i = n;

while (a[i]<a[i-1]) i--;

Arrays.sort(a, i, a.length);

int temp = a[i];

a[i] = a[i-1];

a[i-1] = temp;

return a;

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

while (totalRecords-- > 0) {

int n = sc.nextInt();

sc.nextLine();

int a[] = new int[n+1];

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

a[i] = sc.nextInt();

}

a = HoanVi(a, n);

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

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

}

System.out.println();

}

}

}

**10 Ví dụ với N = 3 ta có kết quả: 321, 312, 231, 213, 132, 123**

public class Bai10 {

private static int a[];

private static int n;

private static boolean isAvailable[];

public static void backTracking(int i) {

if (i == n + 1) {

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

System.out.print(a[j]);

}

System.out.print(" ");

} else {

for (int j = n; j >= 1; j--) {

if (isAvailable[j]) {

a[i] = j;

isAvailable[j] = false;

backTracking(i + 1);

isAvailable[j] = true;

}

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

while (totalRecords-- > 0) {

n = sc.nextInt();

a = new int[n + 1];

isAvailable = new boolean[n + 1];

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

isAvailable[i] = true;

}

backTracking(1);

System.out.println();

}

}

}

**11 Hãy đưa ra tất cả các tập con của xâu ký tự S theo thứ tự tăng dần của các xâu ký tự**

public class Bai11 {

private static int a[];

private static String s;

private static int n;

private static ArrayList<String> results;

public static void backTrack(int i) {

if (i == n+1) {

String result = isIncrease();

if(result != null) results.add(result);

}

else {

for (int j = 0; j <= 1; j++) {

a[i] = j;

backTrack(i+1);

}

}

}

public static String isIncrease(){

int start = 0;

while (start < n && a[start+1] == 0) start ++;

if(start == n) return null;

String temp = ""+s.charAt(start);

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

if(a[i+1]==1){

if(s.charAt(i)<= temp.charAt(temp.length()-1)) return null;

else temp += s.charAt(i);

}

}

return temp;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while (t-- > 0) {

n = sc.nextInt();

sc.nextLine();

results = new ArrayList<>();

s = sc.nextLine();

a = new int[n + 1];

backTrack(1);

Collections.sort(results);

for(String s : results){

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

}

System.out.println();

}

}

}

**12 Với 1 tổ hợp chập K phần tử của N, hãy cho biết tổ hợp tiếp theo sẽ có bao nhiêu phần tử mới**

public class Bai12 {

private static int n;

private static int a[];

private static int k;

public static boolean check(){

int i = 1;

while((i<=k)&&(a[i] == n-k+i)) i++;

if(i==k+1) return true;

return false;

}

public static int getChange(){

if(!check()){

int i = k;

while(a[i] == n-k+i) i--;

int totalDifferent = 0;

int start = a[i]+1;

int end = start + k - i;

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

if(a[j]<start || a[j]>end) totalDifferent++;

}

return totalDifferent;

}

else return k;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

while (totalRecords-- > 0) {

n = sc.nextInt();

k = sc.nextInt();

a = new int[k+1];

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

a[i] = sc.nextInt();

}

System.out.println(getChange());

}

}

}

**13 Bạn hãy lựa chọn các công việc một cách tối ưu sao cho số công việc làm được là nhiều nhất. out 4**

class Todo implements Comparable<Todo>{

private int start;

private int end;

public Todo(int start, int end) {

this.start = start;

this.end = end;

}

public int getStart() {

return start;

}

public void setStart(int start) {

this.start = start;

}

public int getEnd() {

return end;

}

public void setEnd(int end) {

this.end = end;

}

@Override

public int compareTo(Todo o) {

if(this.end==o.end) return 0;

else{

return this.end>o.end?1:-1;

}

}

}

public class No13 {

public static int getIndexOfMinValue(int[] a, int start, int end){

int min = a[start];

int index = start;

for(int i = start; i <= end; i++){

if(a[i]<start){ min = a[i]; index = i;}

}

return index;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int totalRecords = sc.nextInt();

while (totalRecords-- > 0) {

int n = sc.nextInt();

ArrayList<Todo> todos = new ArrayList<>();

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

Todo t = new Todo(sc.nextInt(), sc.nextInt());

todos.add(t);

}

Collections.sort(todos);

int total = 1;

int current = todos.get(0).getEnd();

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

if(todos.get(i+1).getStart()>=current){

current = todos.get(i+1).getEnd();

total ++;

}

}

System.out.println(total);

}

}

}

**14 Hãy tính xem cần ít nhất bao nhiêu thời gian để có thể nối xong N sợi dây?**

public class Bai14 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while (t-- > 0) {

int n = sc.nextInt();

PriorityQueue<Long> ropes = new PriorityQueue<>();

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

ropes.add(sc.nextLong());

}

long totalTime = 0;

while (ropes.size() > 2) {

long first = ropes.poll();

long second = ropes.poll();

long rope = (first + second)%1000000007;

totalTime += rope;

ropes.add(rope);

}

long first = ropes.poll();

long second = ropes.poll();

totalTime = (totalTime + first+second) % (1000000007);

System.out.println(totalTime);

}

}

}

**15 DI CHUYỂN TRONG MÊ CUNG 1 chỉ D-R**

public class No15 {

private static int a[];

private static int n;

private static boolean isAvailable[];

private static int matrix[][];

private static int size;

private static ArrayList<String> results;

public static boolean Findway() {

int x = 1, y = 1;

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

if (a[i] == 1) {

if (y < size && matrix[x][y + 1] == 1) {

y += 1;

}

} else {

if (x < size && matrix[x + 1][y] == 1) {

x += 1;

}

}

}

if (x == size && y == size) {

return true;

}

return false;

}

public static void quayLui(int j) {

if (j <= n) {

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

a[j] = i;

quayLui(j + 1);

}

} else {

if (Findway()) {

String re = "";

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

if (a[i] == 1) {

re += "R";

} else {

re += "D";

}

}

results.add(re);

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while (t-- > 0) {

size = sc.nextInt();

matrix = new int[size + 1][size + 1];

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

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

matrix[i][j] = sc.nextInt();

}

}

if (matrix[1][1] == 0 || matrix[size][size] == 0) {

System.out.println(-1);

} else {

n = 2 \* (size - 1);

a = new int[n + 1];

isAvailable = new boolean[n + 1];

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

isAvailable[i] = true;

}

results = new ArrayList<>();

quayLui(1);

Collections.sort(results);

if (results.size() == 0) {

System.out.print(-1);

} else {

for (String s : results) {

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

}

}

System.out.println();

}

}

}

}

**21  đếm số cách đặt n quân hậu này lên bàn cờ sao cho với 2 quân hậu bất kì, chúng không “ăn” nhau.**

public class Bai21 {

private static int n;

private static int a[];

private static boolean check[];

private static int total;

public static boolean checkCorrect() {

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

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

if ((i - a[i] == j - a[j]) || (i + a[i] == j + a[j])) {

return false;

}

}

}

return true;

}

public static void backTracking(int j) {

if (j == n + 1) {

if (checkCorrect()) {

total++;

}

} else {

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

if (check[i] == true) {

a[j] = i;

check[i] = false;

backTracking(j + 1);

check[i] = true;

}

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while (t-- >0) {

n = sc.nextInt();

a = new int[n + 1];

check = new boolean[n + 1];

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

check[i] = true;

}

total = 0;

backTracking(1);

System.out.println(total);

}

}

}

**22 Nhiệm vụ của bạn là đặt 8 quân hậu lên bàn cờ, sao cho không có 2 quân nào ăn nhau, và số điểm đạt được là lớn nhất.**

public class Bai22 {

private static int n;

private static int a[];

private static int matrix[][];

private static boolean check[];

private static int point;

public static int Dem(){

int p = 0;

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

p += matrix[i][a[i]];

}

return p;

}

public static boolean isCheck() {

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

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

if ((i - a[i] == j - a[j]) || (i + a[i] == j + a[j])) {

return false;

}

}

}

return true;

}

public static void quaylui(int j) {

if (j == n + 1) {

if (isCheck()) {

int p = Dem();

if(p>point) point=p;

}

} else {

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

if (check[i] == true) {

a[j] = i;

check[i] = false;

quaylui(j + 1);

check[i] = true;

}

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while (t-- > 0) {

n = 8;

matrix = new int[9][9];

for(int i = 1; i<=8;i++)

for(int j = 1; j<=8; j++){

matrix[i][j] = sc.nextInt();

}

a = new int[n + 1];

check = new boolean[n + 1];

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

check[i] = true;

}

point = 0;

quaylui(1);

System.out.println(point);

}

}

}

**23  Hãy tìm cách trả ít tờ nhất với số tiền đúng bằng S**

public class Bai23 {

private static int n;

private static int m;

private static int min;

private static int number;

private static int[] money;

public static void find(int x, int i) {

if (x == 0) {

if (number < min) {

min = number;

}

} else {

for (int j = i; j >= 0; j--) {

if (money[j] <= x) {

x = x - money[j];

number++;

find(x, j - 1);

x = x + money[j];

number--;

}

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while (t-- > 0) {

m = sc.nextInt();

n = sc.nextInt();

money = new int[m];

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

money[i] = sc.nextInt();

}

Arrays.sort(money);

min = m+1;

number = 0;

find(n, m - 1);

if (min == m+1) {

System.out.println(-1);

}

else System.out.println(min);

}

}

}

**24 Ví dụ với N = 5 ta có kết quả là: (5), (4, 1), (3, 2), (3, 1, 1), (2, 2, 1), (2, 1, 1, 1), (1, 1, 1, 1, 1**

public class No24 {

private static int n;

private static int[] a;

private static void backTracking(int x, int k, int pre) {

if (k == n + 1) {

if (x == 0) {

System.out.print("(");

int i = 1;

while (a[i]==0) i++;

System.out.print(a[i]);

i++;

while (i <= n) {

if(a[i]>0)System.out.print(" "+a[i]);

i++;

}

System.out.print(") ");

}

} else {

for (int i = pre; i >= 0; i--) {

a[k] = i;

backTracking(x - i, k + 1, i);

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

sc.nextLine();

while (t-- > 0) {

n = sc.nextInt();

a = new int[n + 1];

backTracking(n, 1, n);

System.out.println();

}

}

}

**25 Ví dụ S=”ABC” ta có kết quả {ABC ACB BAC BCA CAB CBA}**

public class Bai25 {

private static String s;

private static int a[];

private static int n;

private static boolean isCheck[];

public static void backTrack(int i) {

if (i == n) {

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

System.out.print(s.charAt(a[j]));

}

System.out.print(" ");

} else {

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

if (isCheck[j]) {

a[i] = j;

isCheck[j] = false;

backTrack(i + 1);

isCheck[j] = true;

}

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

sc.nextLine();

while (t-- > 0) {

s = sc.nextLine();

n = s.length();

a = new int[n];

isCheck = new boolean[n];

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

isCheck[i] = true;

}

backTrack(0);

System.out.println();

}

}

}

**26 Ví dụ với n = 9, k = 3, s = 23, {6, 8, 9} là tập hợp duy nhất thỏa mãn**

public class No26 {

private static int n;

private static int k;

private static int a[];

private static int total;

private static int s;

public static boolean checkTong() {

int t = 0;

for (int i : a) {

t += i;

}

return t == s ? true : false;

}

public static void quayLui(int j) {

if (j == k + 1) {

if (checkTong()) {

total++;

}

} else {

for (int i = a[j-1]+1; i <= n-k+j; i++) {

a[j] = i;

quayLui(j + 1);

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

n = sc.nextInt();

k = sc.nextInt();

s = sc.nextInt();

while (n != 0 || k != 0 || s != 0) {

a = new int[k+1];

total = 0;

quayLui(1);

System.out.println(total);

n = sc.nextInt();

k = sc.nextInt();

s = sc.nextInt();

}

}

}

**31 Hãy thực hiện thuật toán sắp xếp đổi chỗ trực tiếp trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán**

public class Bai31 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int n=sc.nextInt();

int arr[]= new int[n];

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

arr[i]=sc.nextInt();

}

int buoc=1;

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

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

if(arr[j]<arr[i]){

int t= arr[i];

arr[i]=arr[j];

arr[j]=t;

}

}

System.out.print("Buoc "+(i+1)+": ");

for(int j=0;j<n;j++) System.out.print(arr[j]+" ");

System.out.println("");

}

}

}

**32 Hãy thực hiện thuật toán sắp xếp chọn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.**

public class Bai32 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int n=sc.nextInt();

int arr[]= new int[n];

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

arr[i]=sc.nextInt();

}

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

int min=arr[i];

int vtri=i;

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

if(arr[j]<min){

min=arr[j];

vtri=j;

}

}

int m = arr[i];

arr[i]=arr[vtri];

arr[vtri]=m;

System.out.print("Buoc "+(i+1)+": ");

for(int j=0;j<n;j++) System.out.print(arr[j]+" ");

System.out.println("");

}

}

}

**33 Hãy thực hiện thuật toán sắp xếp chèn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán**

public class Bai33 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int n=sc.nextInt();

int m[]= new int[n];

int arr[]= new int[n];

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

arr[i]=sc.nextInt();

}

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

m[i]=arr[i];

Arrays.sort(m,0, i+1);

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

for(int j=0;j<=i;j++) System.out.print(m[j]+" ");

System.out.println("");

}

}

}

**34 thuật toán sắp xếp nổi bọt**

public class Bai34 {

public static boolean check(int arr[]){

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

if(arr[i]>arr[i+1]) return true;

}

return false;

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int n=sc.nextInt();

int arr[]= new int[n];

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

arr[i] = sc.nextInt();

}

int buoc=1;

while(check(arr)){

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

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

int t = arr[i];

arr[i] = arr[i+1];

arr[i+1] = t;

}

}

System.out.print("Buoc "+ buoc +": ");

for(int i=0;i<n;i++) System.out.print(arr[i]+" ");

System.out.println("");

buoc++;

}

}

}

**35 Cho số nguyên dương N và K. Hãy tính NK modulo 109+7.**

public class Bai35 {

static long base =(long)(1e9+7);

public static long LuyThua(int n, long m){

if(m==0) return 1;

if(m==1) return n%base;

long rs= LuyThua(n, m/2);

long a=Tich(rs, rs);

if(m%2==1) return (a\*n)%base;

return a%base;

}

public static long Tich(long a, long b){

if(b==0 ) return 0;

long result= Tich(a,b/2);

if(b%2==1) return (result\*2+a)%base;

return (result\*2)%base;

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc.nextInt();

while(t-->0){

int n=sc.nextInt();

long m=sc.nextLong();

System.out.println(LuyThua(n, m));

}}}

**36 Ví dụ với 2 bước biến đổi, ta có [1] à [1 2 1] à [1 2 1 3 1 2 1].(gấp đôi dãy số)**

public class Bai36 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int solve= sc. nextInt();

while(solve-->0){

int n= sc.nextInt();

long m=sc.nextLong();

int kq=1;

while(m%2==0){

m=(m+1)/2;

kq++;

}

System.out.println(kq);

}

}

}

**38 Nhiệm vụ của bạn là hãy xác định số Fibonaci thứ n. Do đáp số có thể rất lớn, in ra kết quả theo modulo 10^9+7**

public class Bai38 {

public static long base= (long) (1e9+7);

public static long a[][]= new long[2][2];

public static long b[][]= new long[2][2];

public static void solve(long m[][], long n[][]){

long x1=((m[0][0]\*n[0][0])%base+(m[0][1]\*n[1][0])%base)%base;

long x2=((m[0][0]\*n[0][1])%base+(m[0][1]\*n[1][1])%base)%base;

long x3=((m[1][0]\*n[0][0])%base+(m[1][1]\*n[1][0])%base)%base;

long x4=((m[1][0]\*n[0][1])%base+(m[1][1]\*n[1][1])%base)%base;

a[0][0]=x1;

a[0][1]=x2;

a[1][0]=x3;

a[1][1]=x4;

}

public static void find(long f[][],long n){

if (n<=1) return ;

find(f, n/2);

solve(f, f);

if(n%2==1) solve(f, b);

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc. nextInt();

while(t-->0){

Long n= sc.nextLong();

if(n==0) System.out.println(0);

else{

a[0][0]=b[0][0]=1;

a[0][1]=b[0][1]=1;

a[1][0]=b[1][0]=1;

a[1][1]=b[1][1]=0;

find(a, n-1);

System.out.println(a[0][0]);

}

}

}

}

**39 Cho một xâu ký tự str bao gồm nhiều từ trong xâu. Hãy đảo ngược từng từ trong xâu?**

public class Bai39 {

public static StringBuffer inverse(String s){

StringBuffer i= new StringBuffer(s).reverse();

return i;

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc. nextInt();

sc.nextLine();

while(t-->0){

String s= sc.nextLine().trim();

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

for(int i=0;i<arr.length;i++) System.out.print(inverse(arr[i])+" ");

System.out.println("");}}}

**40 Cho một xâu chỉ gồm các kí tự ‘(‘, ‘)’, ‘[‘, ‘]’, ‘{‘, ‘}’. Một dãy ngoặc đúng**

public class Bai40 {

public static boolean test(String s){

if(s.length()%2==1) return false;

Stack st= new Stack();

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

if(s.charAt(i)!='}'&&s.charAt(i)!=']'&&s.charAt(i)!=')') st.push(s.charAt(i));

else if(s.charAt(i)==')'){

if(st.empty()|| String.valueOf(st.pop()).equals("(")==false) return false;

}

else if(s.charAt(i)=='}'){

if(st.empty()|| String.valueOf(st.pop()).equals("{")==false) return false;

}

else if(s.charAt(i)==']'){

if(st.empty()|| String.valueOf(st.pop()).equals("[")==false) return false;

}

}

if (st.empty()) return true;

else return false;

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc. nextInt();

sc.nextLine();

while(t-->0){

String s= sc.nextLine().trim();

if(test(s)) System.out.println("YES");

else System.out.println("NO"); }}}

**44 Hãy viết chương trình chuyển đổi biểu thức biểu diễn dưới dạng tiền tố về dạng hậu tố \*+AB-CD-> AB+CD-\***

public class Bai44 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc.nextInt();

sc.nextLine();

while(t-- >0){

String s= sc.nextLine();

Stack st= new Stack();

for(int i= s.length()-1; i>=0;i--){

if(s.charAt(i)!='+'&&s.charAt(i)!='-'&&s.charAt(i)!='\*'&&s.charAt(i)!='/'){

st.push(s.charAt(i));

}

else {

String str= String.valueOf(s.charAt(i));

String m= String.valueOf(st.pop());

String n= String.valueOf(st.pop());

str= m+n+str;

st.push(str);

}

}

System.out.println(st.pop());

}

}

}

**45 chuyển đổi biểu thức biểu diễn dưới dạng hậu tố về dạng tiền tố.**

public class Bai45 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc. nextInt();

sc.nextLine();

while(t-- >0){

String s=sc.nextLine();

Stack st= new Stack();

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

if(s.charAt(i)!='+'&&s.charAt(i)!='- '&&s.charAt(i)!='\*'&&s.charAt(i)!='/'){

st.push(s.charAt(i));

}

else {

String str= String.valueOf(s.charAt(i));

String m= String.valueOf(st.pop());

String n= String.valueOf(st.pop());

str= str+n+m;

st.push(str);

}

}

System.out.println(st.pop());

}

}

}

**46 Hãy viết chương trình chuyển tính toán giá trị của biểu thức hậu tố 231\*+9–=-4**

public class Bai46 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc. nextInt();

sc.nextLine();

while(t-- >0){

String s=sc.nextLine();

Stack st= new Stack();

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

if(s.charAt(i)!='+'&&s.charAt(i)!='-'&&s.charAt(i)!='\*'&&s.charAt(i)!='/'){

st.push(s.charAt(i));

}

else {

int c=0;

int a= Integer.parseInt(String.valueOf(st.pop()));

int b= Integer.parseInt(String.valueOf(st.pop()));

if(s.charAt(i)=='+') {

c= a+b;

}

if(s.charAt(i)=='-') {

c= b-a;

}

if(s.charAt(i)=='\*') {

c= b\*a;

}

if(s.charAt(i)=='/') {

c= (int)b/a;

}

st.push(c);

}

}

System.out.println(String.valueOf(st.pop()));

}

}

}

**47 tính toán giá trị của biểu thức tiền tố**

public class Bai47 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc.nextInt();

sc.nextLine();

while(t-->0){

String s= sc.nextLine();

Stack st= new Stack();

for(int i= s.length()-1; i>=0;i--){

if(s.charAt(i)!='+'&&s.charAt(i)!='-'&&s.charAt(i)!='\*'&&s.charAt(i)!='/'){

st.push(s.charAt(i));

}

else {

int c= 0;

int a= Integer.parseInt(String.valueOf(st.pop()));

int b= Integer.parseInt(String.valueOf(st.pop()));

if(s.charAt(i)=='+') {

c= a+b;

}

if(s.charAt(i)=='-') {

c= a-b;

}

if(s.charAt(i)=='\*') {

c= b\*a;

}

if(s.charAt(i)=='/') {

c= (int)a/b;

}

st.push(c);

}

}

System.out.println(st.pop());

}

}

}

**49 Cho dãy ký tự S chỉ bao gồm các ký tự I hoặc D. Ký tự I được hiểu là tăng (Increasing) ký tự D được hiểu là giảm (Decreasing). Sử dụng các số từ 1 đến 9, hãy đưa ra số nhỏ nhất được đoán nhận từ S. Chú ý, các số không được phép lặp lại**

public class Bai49 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc.nextInt();

while(t-- >0){

int a= sc.nextInt();

long b[]= new long[a];

for(int i=0;i<a;i++) b[i]= sc.nextLong();

long dientichS=0;

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

long cao= b[i];

long rong= 1;

int n=i-1;

while(n>=0&& b[n]>=b[i]) {

rong++;

n--;

}

n=i+1;

while(n<a&&b[n]>=b[i]){

rong++;

n++;

}

long dt= rong\*cao;

if(dt>dientichS) dientichS= dt;

}

System.out.println(dientichS);

}

}

}

**50 Hãy in ra tất cả các số nhị phân từ 1 đến n**

public class Bai50 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int sd[]=new int[10];

int test= sc.nextInt();

sc.nextLine();

while(test-->0){

for(int i=1;i<=9;i++) sd[i]=0;

String s= sc.nextLine().trim();

int n=s.length();

int a[] = new int[n+1];

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

int x= 1;

while(sd[x]==1) x++;

a[i]=x;

sd[x]=1;

if(s.charAt(i)=='D'){

int t=i;

sd[x]=0;

x=a[i]+1;

while(t>=1 && s.charAt(t-1)=='D') {

t--;

}

for(int j=i;j>t;j--) {

a[j]=a[j-1];

}

while(sd[x]==1) x++;

a[t]=x;

sd[x]=1;

}

}

int x=1;

while(sd[x]==1) x++;

a[n]=x;

for(int i=0;i<=n;i++) System.out.print(a[i]);

System.out.println("");

}

}

}

**51 có bao nhiêu số BDN không lớn hơn N. Ví dụ N=100 ta có 4 số BDN bao gồm các số: 1, 10, 11, 100**

public class Bai51 {

public static String sinhNP(String s){

int n= s.length();

int x= n-1;

while(x>=0&& s.charAt(x)=='1') x--;

String s1="";

if (x==-1){

s1+="1";

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

}

else {

for(int i=0;i<x;i++) s1+=s.charAt(i);

s1+="1";

for(int i=x+1;i<s.length();i++) s1+="0";

}

return s1;

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc.nextInt();

while(t-- >0){

int n= sc.nextInt();

String s="0";

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

s=sinhNP(s);

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

}

System.out.println("");

}

}

}

**52 Số BDN của N là số P =M´N sao cho P là số BDN. Cho số tự nhiên N (N<1000), hãy tìm số BDN nhỏ nhất của N**

public class Bai52 {

public static long y;

public static long numofBDN(String s){

int i=0;

int n= s.length();

while(i<n&&s.charAt(i)=='0') i++;

if(i==n) return 0;

else{

String str="";

for(int j=i;j<n;j++) str+=s.charAt(j);

n=str.length();

long sl=(long) Math.pow(2, n-1);

int j=0;

if(str.charAt(j)>'1') {

sl\*=2;

y+=sl;

if(j==str.length()-1) y++;

return y-1;

}

else if(str.charAt(j)=='1'){

String s2="";

for(int j1=1;j1<str.length();j1++) s2+=str.charAt(j1);

return y+ sl+ numofBDN(s2);

}

}

return y;

}

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

int t= sc.nextInt();

sc.nextLine();

while(t-->0){

String s= sc.nextLine().trim();

y=0;

System.out.println(numofBDN(s));

}

}

}

**53 Hãy tìm số nguyên dương X nhỏ nhất được tạo bởi số 9 và số 0 chia hết cho N. Ví dụ với N = 5 ta sẽ tìm ra X = 90**

public class Bai53 {

public static long sinhNP(long s){

String str= String.valueOf(s);

int n= str.length();

int i=n-1;

while(i>0&& str.charAt(i)=='1') i--;

if(i==0){

String s1="1";

for(int j=0;j<str.length();j++) s1+="0";

s= Long.parseLong(s1);

}

else {

String s1="";

for(int j=0;j<i;j++) s1+=str.charAt(j);

s1+="1";

for(int j=i+1;j<str.length();j++) s1+="0";

s= Long.parseLong(s1);

}

return s;

}

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

int t= sc.nextInt();

sc.nextLine();

while(t-->0){

int n=sc.nextInt();

if(n==0) System.out.println(0);

else{

Long s1=(long)1;

while(s1 %n!=0){

s1=sinhNP(s1);

}

System.out.println(s1);

}

}

}

}

**54 Hãy dịch chuyển S thành T sao cho số lần thực hiện các thao tác (a), (b) là ít nhất**

public class Bai54 {

public static String sinhNP(String s){

int n= s.length();

int x= n-1;

while(x>=0&& s.charAt(x)=='9') x--;

String s1="";

if (x==-1){

s1+="9";

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

}

else {

for(int i=0;i<x;i++) s1+=s.charAt(i);

s1+="9";

for(int i=x+1;i<s.length();i++) s1+="0";

}

return s1;

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc.nextInt();

while(t-- >0){

int n= sc.nextInt();

String s="0";

while(Integer.parseInt(sinhNP(s))%n!=0){

s=sinhNP(s);

}

System.out.println(Integer.parseInt(sinhNP(s)));

}

}

}

**56 Hãy tìm xâu con chung dài nhất của 2 xâu này**

public class Bai56 {

public static int calcu(int s, int m){

if(s>=m) return s-m;

int dem=0;

if(m%2==1){

dem=1;

m++;

}

return (dem+1+ calcu(s, m/2));

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t= sc.nextInt();

while(t-->0){

int s= sc.nextInt();

int m=sc.nextInt();

int dem=0;

if(s>=m) dem=s-m;

else{

dem=calcu(s,m);

}

System.out.println(dem);

}

}

}

**61 Cho xâu ký tự S. Nhiệm vụ của bạn là tìm độ dài dãy con lặp lại dài nhất trong S**

public class Bai61 {

private static int check(String s1, String s2) {

int kq = 0;

int n = s1.length();

int m = s2.length();

int arr[][] = new int[1001][1001];

s1 = " " + s1;

s2 = " " + s2;

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

for (int j = 0; j <= s2.length(); j++) {

arr[i][j] = 0;

}

}

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

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

if (s1.charAt(i) == s2.charAt(j)) {

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

} else {

arr[i][j] = Math.max(arr[i][j - 1], arr[i - 1][j]);

}

kq = Math.max(kq, arr[i][j]);

}

}

return kq;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t;

t = sc.nextInt();

String s1, s2;

sc.nextLine();

while (t-- > 0) {

s1 = sc.nextLine();

s2 = sc.nextLine();

System.out.println(check(s1, s2));

}

}

}

**63 Hãy cho biết dãy con tăng dài nhất của dãy này có bao nhiêu phần tử?**

public class **DayConTangDaiNhat** {

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

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int [] arr = new int [n];

int [] f = new int [n];

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

arr[i] = sc.nextInt();

}

int result = 1;

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

f[i] = 0;

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

if(arr[j] < arr[i]){

f[i] = Math.max(f[i], f[j]);

}

}

f[i] += 1;

result = Math.max(result, f[i]);

}

System.out.println(result); }}

**64 Nhiệm vụ của bạn là tìm tổng lớn nhất của dãy con được sắp theo thứ tự tăng dần của dãy A[]**

public class Bai64 {

private static int check(int[] a, int n) {

int kq = 0;

int arr[] = new int[n];

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

arr[i] = 1;

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

if(a[j] < a[i]) {

arr[i] = Math.max(arr[i], arr[j]+1);

}

}

kq = Math.max(kq,arr[i]);

}

return kq;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n;

n = sc.nextInt();

int a[] = new int[n];

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

a[i] = sc.nextInt();

}

System.out.println(check(a, n));

}

}

65 Nhiệm vụ của bạn **là sắp xếp lại mảng số với số lượng bước là ít nhất**

public class Bai65 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

sc.nextLine();

while (t-- > 0) {

int n = sc.nextInt();

int arr[] = new int[n + 5];

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

arr[i] = sc.nextInt();

}

int kq = 0;

int dp[] = new int[1001];

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

dp[i] = arr[i];

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

if (arr[j] < arr[i]) dp[i] = Math.max(dp[i], dp[j] + arr[i]);

kq = Math.max(dp[i], kq);

}

}

System.out.println(kq);

}

}

}

**66 Một con ếch có thể nhảy 1, 2, 3 bước**

public class Bai66 {

private static int check(int[] a) {

int n = a.length;

int kq = 0;

int F[] = new int[1005];

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

F[i] = 1;

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

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

F[i] = Math.max(F[i], F[j] +1);

}

}

kq = Math.max(kq, F[i]);

}

return n - kq;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t, n;

t = sc.nextInt();

while(t-- >0) {

n = sc.nextInt();

int a[] = new int[n];

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

a[i] = sc.nextInt();

}

System.out.println(check(a));

}

}

}

**67 Cho 2 số nguyên n, k. Bạn hãy tính C(n, k) modulo 109+7.**

public class Bai67 {

private static long check(int n) {

long F[] = new long[55];

F[0] = 1;F[1] = 1;F[2] = 2;

for (int i = 3; i <= n; i++) {

for (int j = 1; j <= 3; j++) {

F[i] += F[i - j];

}

}

return F[n];

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t, n;

t = sc.nextInt();

while (t-- > 0) {

n = sc.nextInt();

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

}

}

}

**68 Một chiếc cầu thang có N bậc**

public class Bai68 {

static int m = 1000000007;

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t;

t = sc.nextInt();

while (t-- > 0) {

int n, k;

n = sc.nextInt();

k = sc.nextInt();

int C[][] = new int[1005][1005];

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

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

if(j == 0 || j == i) {

C[i][j] = 1;

}

else {

C[i][j] = (C[i-1][j] + C[i-1][j-1]) %m;

}

}

} System.out.println(C[n][k]);}}}

**69 Hãy tìm xâu con đối xứng dài nhất của S**

public class Bai69 {

static int m = 1000000007;

private static long check(int n, int k) {

long arr[] = new long[1000005];

arr[0] = 1;

arr[1] = 1;

if(k == 1) arr[2] = 1;

else arr[2] = 2;

for(int i = 3; i <=n; i++) {

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

if(i - j >=0) {

arr[i] += arr[i-j];

arr[i] %= m;

}

}

}

return arr[n];

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t, n, k;

t = sc.nextInt();

while(t-- >0) {

n = sc.nextInt();

k = sc.nextInt();

System.out.println(check(n, k));

}

}

}

**72 Viết ký tự giống nhau .Ví dụ với N = 9, X =1, Y = 2, Z =1 ta có kết quả là 5 bằng cách thực hiện: insert, insert, copying, copying, inser**

public class KyTuGiongNhau {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int test = sc.nextInt();

while (test-- > 0) {

int n = sc.nextInt();

int in = sc.nextInt();

int re = sc.nextInt();

int co = sc.nextInt();

int dp[] = new int[n + 5];

dp[0] = 0;

dp[1] = in;

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

if (i % 2 == 1) {

dp[i] = Math.min(dp[i - 1] + in, dp[(i + 1) / 2] + re + co);

} else {

dp[i] = Math.min(dp[i - 1] + in, dp[i / 2] + co);

}

}

System.out.println(dp[n]);

}

}

}

**73 Một bản tin M đã mã hóa bí mật thành các con số theo ánh xạ như sau: ‘A’->1, ‘B’->2, .., ‘Z’->26**

public class MaHoa {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int test = sc.nextInt();

sc.nextLine();

while(test-- > 0){

String s = sc.nextLine();

if(s.charAt(0) == '0'){

System.out.println(0);

continue;

}

int n = s.length();

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

dp[0] = 1;

dp[1] = 1;

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

dp[i] = 0;

if(s.charAt(i-1) > '0'){

dp[i] = dp[i-1];

}

if(s.charAt(i-2) == '1' || (s.charAt(i-2) == '2' && s.charAt(i-1) < '7'))

dp[i] += dp[i-2];

}

System.out.println(dp[n]);

}

}

}

**74 Nhiệm vụ của bạn là tìm số lượng ít nhất các số nhỏ hơn N mà có tổng bình phương bằng N**

public class TongBinhPhuong {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int test = sc.nextInt();

while(test-- > 0){

int n = sc.nextInt();

int arr [] = new int[n+5];

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

arr[i] = i;

}

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

for(int j = 1; j <= Math.sqrt(i); j++)

arr[i] = Math.min(arr[i], arr[i-j\*j]+1);

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

}

}

}

**79 Bạn hãy tìm hình chữ nhật lớn nhất bị che phủ bởi các cột?**

**ChuyenDanhSachCanhSangDanhSachKe**

public class **ChuyenDanhSachCanhSangDanhSachKe** {

static Scanner sc = new Scanner(System.in);

public static void input (ArrayList<Integer>[] map) {

int a = sc.nextInt();

int b = sc.nextInt();

map[a].add(b);

map[b].add(a);

}

public static void ShowMap (ArrayList<Integer>[] map) {

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

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

for (int value : map[i]) {

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

}

System.out.println();

}

}

public static int ham () {

try {

} catch (Exception e) {

e.printStackTrace();

return 1;

}

return 0;

}

public static void main(String[] args) {

if (ham() == 1) return;

int loop = sc.nextInt();

while (loop-->0) {

int v = sc.nextInt();

int e = sc.nextInt();

ArrayList<Integer>[] map = new ArrayList[v+1];

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

map[i] = new ArrayList<>();

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

input(map);

}

ShowMap(map);

}}}

###########################################

**ChuyenDanhSachKeSangDanhSachCanh**

public class ChuyenDanhSachKeSangDanhSachCanh {

static Scanner sc = new Scanner(System.in);

public static void input (ArrayList<Integer>[] map) {

int a = sc.nextInt();

int b = sc.nextInt();

map[a].add(b);

map[b].add(a);

}

public static void ShowEdge (ArrayList<Integer>[] map) {

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

for (int value : map[i]) {

if (i < value)

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

}

}

}

public static void main(String[] args) {

int v = Integer.valueOf(sc.nextLine());

ArrayList<Integer>[] map = new ArrayList[v+1];

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

map[j] = new ArrayList<>();

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

String line = sc.nextLine();

Scanner numbers = new Scanner(line);

while (numbers.hasNext()) {

map[i].add(numbers.nextInt());

}

}

ShowEdge(map);

}

}

####################################################

public class **DayConCoTongBangS** {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int test = in.nextInt();

while (test-- > 0) {

int n = in.nextInt();

int s = in.nextInt();

int a;

int dp[] = new int[1001];

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

a = in.nextInt();

dp[a] = 1;

for (int j = s; j >= a; j--) {

if (dp[j - a] == 1) {

dp[j] = 1;

}

}

}

if (dp[s] == 1) {

System.out.println("YES");

} else {

System.out.println("NO");

}

}

}

}

#######################################

public class **CaiTui** {

static int F[][]=new int[1005][1005];

static int a[]=new int[1005];

static int c[]=new int [1005];

static int n,v;

public static int tinh(){

for(int i=0;i<=n;i++) F[i][0]=0;

for(int i=0;i<=v;i++)F[0][i]=0;

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

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

F[i][j]=F[i-1][j];

if(j>=a[i]){

F[i][j]=Math.max (F[i-1][j-a[i]]+c[i],F[i-1][j]);

}

}

}return F[n][v];

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int t=sc.nextInt();

while(t-->0){

n=sc.nextInt();

v=sc.nextInt();

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

a[i]=sc.nextInt();

}

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

c[i]=sc.nextInt();

}System.out.println(tinh());

}

}

}

#############################################

public class **BieuDienDoThiCoHuong** {

static Scanner sc = new Scanner(System.in);

public static void input (ArrayList<Integer>[] map) {

int a = sc.nextInt();

int b = sc.nextInt();

map[a].add(b);

}

public static void ShowMap (ArrayList<Integer>[] map) {

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

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

for (int value : map[i]) {

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

}

System.out.println();

}

}

public static void main(String[] args) throws IOException {

int loop = sc.nextInt();

while (loop-->0) {

int v = sc.nextInt();

int e = sc.nextInt();

ArrayList<Integer>[] map = new ArrayList[v+1];

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

map[i] = new ArrayList<>();

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

input(map);

}

ShowMap(map);

}

}

}

########################################################

public class **Bfs** {

public static Scanner sc = new Scanner(System.in);

public static boolean cx[] = new boolean[1001];

public static int a[][] = new int[1001][1001];

public static int count, start, m, n;

public static void main(String[] args) {

int test = sc.nextInt();

while (test-- > 0) {

init();

BFS(start);

System.out.println("");

}

}

public static void init() {

n = sc.nextInt();

m = sc.nextInt();

start = sc.nextInt();

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

cx[i] = true;

Arrays.fill(a[i], 0);

}

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

int x = sc.nextInt();

int y = sc.nextInt();

a[x][y] = 1;

a[y][x] = 1;

}

}

public static void BFS(int u) {

Queue<Integer> qu = new LinkedList();

qu.add(u);

cx[u] = false;

while (!qu.isEmpty()) {

int v = qu.peek();

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

qu.poll();

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

if (cx[i] && a[v][i] == 1) {

qu.add(i);

cx[i] = false;

}

}

}

}

}

#######################

public class **Dfs** {

public static Scanner sc = new Scanner(System.in);

public static boolean cx[] = new boolean[1001];

public static int a[][] = new int[1001][1001];

public static int start, m, n;

public static void main(String[] args) {

int test = sc.nextInt();

while (test-- > 0) {

init();

DFS(start);

System.out.println("");

}

}

public static void init() {

n = sc.nextInt();

m = sc.nextInt();

start = sc.nextInt();

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

cx[i] = true;

Arrays.fill(a[i], 0);

}

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

int x = sc.nextInt();

int y = sc.nextInt();

a[x][y] = 1;

a[y][x] = 1;

}

}

public static void DFS(int u){

Stack st = new Stack();

st.push(u);

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

cx[u] = false;

while(!st.isEmpty()){

int v = (int) st.peek();

st.pop();

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

if(cx[i] && a[v][i] == 1){

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

cx[i] = false;

st.push(v);

st.push(i);

break;

}

}

}

}

}