To find the element where pattern starts increasing::

#include<stdio.h>

void main()

{

int n,a[100],i,j,k;

scanf("%d",&n);

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

{

scanf("%d",&a[i]);

}

j=1;

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

{

k=a[i]-a[j];

if(k<0)

{

printf("%d %dindexposition",a[i],i);

break;

}

j++;

}

}

Longest palindrome substring::

import java.util.\*;

public class Main {

static void printSubStr(String str1, int l, int h) {

System.out.println(str1.substring(l, h + 1));

}

static int longPalSubstr(String str1) {

int n = str1.length();

boolean table[][] = new boolean[n][n];

int mLength = 1;

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

table[i][i] = true;

int strt = 0;

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

if (str1.charAt(i) == str1.charAt(i + 1)) {

table[i][i + 1] = true;

strt = i;

mLength = 2;

}

}

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

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

int j = i + k - 1;

if (table[i + 1][j - 1] && str1.charAt(i) == str1.charAt(j)) {

table[i][j] = true;

if (k > mLength) {

strt = i;

mLength = k;

}

}

}

}

System.out.print("The longest palindrome substring in the given string is; ");

printSubStr(str1, strt, strt + mLength - 1);

return mLength;

}

public static void main(String[] args) {

String str1 ;

Scanner s=new Scanner(System.in);

Str1=s.next();

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

System.out.println("The length of the palindromic substring is: " + longPalSubstr(str1));

}

}