Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Отчет по лабораторной работе №4

на тему: «Реализация стека и дека»

Выполнил: студент группы БВТ1903

Белов Сергей Павлович

Проверил:

Павликов Артём Евгеньевич

# **Треугольник с максимальным периметром**

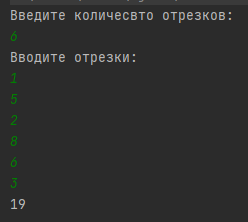
Задача:

Массив A состоит из целых положительных чисел ­ длин отрезков. Составьте из трех отрез­ков такой треугольник, чтобы его периметр был максимально возможным. Если невозможно составить треугольник с положительной площадью ­ функция возвращает 0.

Листинг:

public class Triangle {  
 public static void main (String[] args){  
 Scanner s = new Scanner(System.*in*);  
 System.*out*.println("Введите количесвто отрезков:");  
 int n = s.nextInt();  
 int [] arr = new int [n];  
 System.*out*.println("Вводите отрезки:");  
 for (int i=0;i<n;i++){  
 arr[i]=s.nextInt();  
 }  
 int max = 0; int temp = 0;  
 for (int i=0;i<n-2;i++){  
 for(int j=i+1;j<n-1;j++){  
 for (int k=j+1;k<n;k++){  
 if((arr[i]<(arr[j]+arr[k]))&&(arr[j]<(arr[i]+arr[k]))&&(arr[k]<(arr[j]+arr[i]))){  
 temp=arr[i]+arr[j]+arr[k];  
 if(temp>max){  
 max=temp;  
 }  
 }else continue;  
  
 }  
 }  
 }  
 System.*out*.println(max);  
 }  
}

Результаты работы программы:



# **Максимальное число**

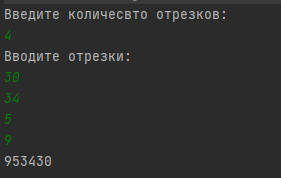
Задача:

Дан массив неотрицательных целых чисел nums. Расположите их в таком порядке, чтобы вместе они образовали максимально возможное число.

Листинг:

public class Max\_Value {//4,30,34,5,9  
 public static void main (String[] args) {  
 Scanner s = new Scanner(System.*in*);  
 System.*out*.println("Введите количесвто отрезков:");  
 int n = s.nextInt();  
 int[] arr = new int[n];  
 System.*out*.println("Вводите отрезки:");  
 for (int i = 0; i < n; i++) {  
 arr[i] = s.nextInt();  
 }  
 String [] str = new String[n];  
 for (int i=0;i<n;i++){  
 str[i]=String.*valueOf*(arr[i]);  
 }  
 String res = "";  
 while (str.length!=0){  
 res+=*Max*(str);  
 str =*Delete*(str,*Max*(str));  
 }  
 System.*out*.println(res);  
  
 }  
 public static String Max(String[] s){  
 String max="";  
 int maxim = 0;  
 for(int i=0;i<s.length;i++){  
 if(Integer.*parseInt* (String.*valueOf*(s[i].charAt(0)))>maxim){  
 max=s[i];  
 }  
 }return max;  
 }  
 public static String[] Delete (String[] s, String max){  
 String[] s1 = new String[s.length-1];  
 for (int i=0;i<s.length;i++){  
 if(s[i]!=max) s1[i]=s[i];  
 }return s1;  
 }  
}

Результаты работы программы:



# **Сортировка диагоналей в матрице**

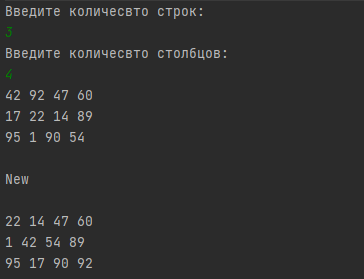
Задача:

Дана матрица mat размером m \* n, значения ­ целочисленные. Напишите функцию, сор­тирующую каждую диагональ матрицы по возрастанию и возвращающую получившуюся матрицу.

Листинг:

public class Diagonal {  
 static int *n*;  
 static int *m*;  
 static int[][] *arr*;  
 static int[][] *new\_arr*;  
 public static void main (String[] args) {  
 Scanner s = new Scanner(System.*in*);  
 System.*out*.println("Введите количесвто строк:");  
 int n1 = s.nextInt();  
 System.*out*.println("Введите количесвто столбцов:");  
 int m1 = s.nextInt();  
 *n*=n1;  
 *m*=m1;  
 *arr* =new int[*n*][*m*];  
 *new\_arr* = new int[*n*][*m*];  
  
 //заполняем массив строками  
 for (int i = 0; i < *n*; i++) {  
 for (int j=0;j<*m*;j++){  
 *arr*[i][j] = (int)((Math.*random*() \* ((100 + 1))));  
 //arr[i][j]=s.nextInt();  
 }  
 }  
 //выводим массив  
 for (int i = 0; i < *n*; i++) {  
 for (int j=0;j<*m*;j++){  
 System.*out*.print(*arr*[i][j]+" ");  
 }  
 System.*out*.println();  
 }  
  
 for (int i = 0; i < *n*; i++) {  
 for (int j=0;j<*m*;j++){  
 *Diag*(i,j);  
 }  
 }  
  
 //отделяем место и выводим отсортированный массив  
 System.*out*.println();  
 System.*out*.println("New");  
 System.*out*.println();  
 for (int i = 0; i < *n*; i++) {  
 for (int j=0;j<*m*;j++){  
 System.*out*.print(*new\_arr*[i][j]+" ");  
 }  
 System.*out*.println();  
 }  
  
  
 }  
 public static void Diag (int string, int column) {  
 Deque <Integer> deque = new LinkedList<>();  
 int count = 0;  
 int i=string; int j=column;  
 while (i< *n*&&j< *m*){  
 if(*arr*[i][j]>0) {  
 deque.addLast(*arr*[i][j]);  
 count++;  
 *arr*[i][j] = -1;  
 i++;j++;  
 }else{break;}  
 }if(deque.isEmpty()!=true) {  
 int[] diagonalArr = new int[count];  
 for (int k = 0; k < count; k++) {  
 diagonalArr[k] = deque.pollFirst();  
 }  
 Arrays.*sort*(diagonalArr);  
 for (int k = 0; k < count; k++) {  
 deque.addLast(diagonalArr[k]);  
 }  
 i = string;  
 j = column;  
 while (i < *n* && j < *m*) {  
 *new\_arr*[i][j] = deque.pollFirst();  
 i++;  
 j++;  
 }  
 }  
 }  
  
}

Результаты работы программы:



# **Объединение отрезков**

Задача

Дан массив отрезков intervals, в котором intervals[i] = [starti, endi], некоторые отрезки могут пересекаться. Напишите функцию, которая объединяет все пересекающиеся отрезки в один и возвращает новый массив непересекающихся отрезков.

Листинг:

public class Interval {  
 public static void main (String [] args){  
 String input="";  
 System.*out*.println("Ввод:");  
 Scanner s = new Scanner(System.*in*);  
 input=s.nextLine();  
 int [] array = *Search\_interval*(input);  
 int counter = 0;  
 Deque <Integer> deque = new LinkedList<>();  
 for (int i=0;i<(array.length)-2;i++){  
 if (array[i+1]>=array[i+2]&&array[i+1]<=array[i+3]){  
 deque.addLast(array[i]);  
 deque.addLast(array[i+3]);  
 i+=3;  
 counter+=2;  
 }  
 else{  
 deque.addLast(array[i]);  
 deque.addLast(array[i+1]);  
 counter+=2;  
 i+=2;  
 }  
 }  
 //костыль  
 if(array[array.length-1]!= deque.peekLast()){  
 deque.addLast(array[array.length-2]);  
 deque.addLast(array[array.length-1]);  
 counter+=2;  
 }  
 *Print*(*Deq\_to\_int*(deque,counter));  
 //System.out.println(Arrays.toString(Deq\_to\_int(deque,counter)));  
 //System.out.println(Arrays.toString(array));  
 }  
 public static int [] Search\_interval(String string){  
 Deque <Integer> deque = new LinkedList<>();  
 int count = 0;  
 String temp = "";  
 for (int i=1;i<string.length()-1;i++){  
 if (string.charAt(i)=='['){  
 while (string.charAt(i)!=']'){  
 if(string.charAt(i)>='0'&&string.charAt(i)<='9'){  
 temp+=string.charAt(i);  
 i++;  
 }else if (string.charAt(i)==','){  
 deque.addLast(Integer.*parseInt*(temp));  
 count++;  
 i++;  
 temp="";  
 }else i++;  
 }  
 deque.addLast(Integer.*parseInt*(temp));  
 count++;  
 //i++;  
 temp="";  
 }  
  
 }  
 return *Deq\_to\_int*(deque,count);  
 }  
 public static int [] Deq\_to\_int (Deque <Integer> deque, int count){  
 int [] arr = new int [count];  
 for (int i=0;i<count;i++){  
 arr[i]=deque.pollFirst();  
 }  
 return arr;  
 }  
 public static void Print (int []arr){  
 String str = "[";  
 for (int i=0;i< arr.length;i++){  
 if (i%2==0){  
 str+="["+arr[i]+",";  
 }  
 else {str+=arr[i]+"], ";}  
 }  
 str=str.substring(0, str.length() - 2);  
 str+="]";  
 System.*out*.println(str);  
 }  
}

Результаты работы программы:

