Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 10

По дисциплине «Основы алгоритмизации и программирования»

На тему «Одномерные массивы»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условие задачи** |
| --- | --- |
| 9 | 1. Найти в массиве элемент, наиболее близкий к среднему арифметическому суммы его элементов.  2. Задано два массива. Найти наибольшие среди элементов первого массива, которые не входят во второй массив. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>//директива, необходимая для функции time  using namespace std;  int main()  {  setlocale(LC\_CTYPE, "Russian");  int n, x, sum = 0;  const int max\_n = 32000;//ограничиваем максимальный размер  cout << "Введите размер массива: ";  cin >> n;  if (max\_n < n) {//проверка на введеный пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  int array[max\_n];//объявляем массив  srand((unsigned)time(NULL));//с помощью функции srand(seed of random) генерируем новые числа при каждом запуске,функция time(NULL),возвращает текущее время в секундах  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы массива, наш произвольный размер достигается за счёт того, что мы ведем наш цикл ввода и остальные лишь до указанного размера, тем самым игнорируя остальные элементы. Не задавая им значения, в функции int main они примут значение 0  {  array[i]=rand()%100;//генерируем по одному элементу массива за шаг  cout << array[i] << " ";//выводим элементы массива для возможности их анализа  sum += array[i];//сразу считаем сумму всех элементов для среднего арифметического  }  x = array[0];//для поиска нужного числа присваиваем переменной x значение 1-ого элемента массива, относительного которого будут вести дальнейшие сравнения до того, пока не будет найден более подходящий кандидат  sum /= n;//находим среднее арифметическое  cout<<endl << "Среднее арифметическое элементов массива: " << sum << endl;//выводим его на экран, чтобы упростить анализ работоспособность программы  for (int i = 0; i < n; i++)  {  if (abs(sum - array[i]) < abs(sum - x)) { x = array[i]; }//сравниваем каждый элемент массива. Чем меньше разница по модулю, тем ближе находиться данные числа на координатной прямой  }  cout << x;//выводим искомое число  } |
| **Результат программы** |
|  |
| **Блок-схема** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>//директива, необходимая для функции time  using namespace std;  int main()  {  setlocale(LC\_CTYPE, "Russian");  int n, k, max = 0;  const int max\_n = 32000;  cout << "Введите размер первого массива: ";  cin >> n;  cout << "Введите размер второго массива: ";  cin >> k;  srand((unsigned)time(NULL));  if (max\_n < n or max\_n < k) {//проверка на введеный пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  int arr\_a[max\_n], arr\_b[max\_n];//объявляем массивы  cout << endl << "Первый массив" << endl << n << endl;  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы первого массива  {  arr\_a[i] = rand() % 100;//генерируем по одному элементу первого массива за шаг в диапозоне до 100  cout << arr\_a[i] << " ";//выводим элементы массива для возможности их анализа  }  cout << endl << endl << "Второй массив" << endl << k << endl;  for (int i = 0; i < k; i++)//с помощью цикла for вводим элементы второго массива  {  arr\_b[i] = rand() % 100;//генерируем по одному элементу второго массива за шаг в диапозоне до 100  cout << arr\_b[i] << " ";//выводим элементы массива для возможности их анализа  for (int j = 0; j < n; j++) {//проверяем находиться ли у нас число в 1 массиве и во 2  if (arr\_a[j] == arr\_b[i]) { arr\_a[j] = -1; }//если число повторяется,то мы убираем его из диапазона 0-99  }  }  for (int i = 0; i < n; i++)  {  if (max < arr\_a[i]) {//идем по первому массиву в поисках максимального элемента  max = arr\_a[i];  }  }  cout << endl << endl << max;//выводим искомое число  } |
| **Результат программы** |
| **Элементы 95 и 93 были исключены** |

Дополнительные задания

6.Найти непрерывную последовательность положительных чисел, сумма элементов в которой максимальная. Максимальный размер массива 100 элементов. Диапазон значений от -100 до 100.

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>  using namespace std;  int main()  {  setlocale(LC\_CTYPE, "Russian");  int n, max=0, sum = 0,ind=0,ind\_max=0,counter=0,counter\_max=0;  const int max\_n = 100;  cout << "Введите размер массива: ";  cin >> n;  if (max\_n < n) {  cout << "Превышение максимального размера массива!";  return 0;  }  int array[max\_n];  srand((unsigned)time(NULL));  for (int i = 0; i < n; i++)  {  array[i]=(rand()%200)-100;//генерируем числа в диапазоне [-100;100]  cout << array[i] << " ";    if(array[i]>0) {//если элемент положительный,то добовляем его к переменной sum для посчета суммы всех последовательных положительных,а также ведем счет сколько элементов находиться в последовательности  if(sum==0){//если элемент является первым в последовательности,то ещё запоминаем его индекс  sum += array[i];  ind = i;  counter++;  }  else { sum += array[i]; counter++; }  }  if(array[i] <= 0 or (array[i] > 0 and i==n-1)) {//как только мы встречаем не положительный элемент,то заканчиваем считать сумму последовательности или последний элемент является положительным  if (sum > max) {  max = sum;// если полученная сумма больше встреченной до этого,то записываем ее как максимальную  ind\_max = ind;//запоминаем индекс первого элемента максимальной последовательности  counter\_max = counter;//запоминаем кол-во элементов в максимальной последовательности  }  sum = 0;//начинаем счет следующей суммы заново  counter = 0;//начинаем счёт кол-ва элементов заново  }    }  counter\_max += ind\_max;//вычисляем индекс (последнений элемент + 1) последовательности  cout << endl << "Искомый отрезок:" << endl;  for (int j = ind\_max; j < counter\_max; j++) {//начинаем с первого элемента максимальной последовательности и до последнего элемента  cout << array[j] << " ";//выводим каждый элемент  }  cout << endl<<"Сумма элементов отрезка: " << max;//выводим сумму данного отрезка для упрощения анализа результатов программы  } |
| **Результат программы** |
|  |

5.В массиве А, размером 2n + 1, не содержащем одинаковых элементов, найти средний по величине элемент, т. е. такой, при котором в массиве А ровно n элементов меньше его и столько же элементов больше его. Массив А сохранить (не сортировать), дополнительных массивов не использовать.

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>//директива, необходимая для функции time  using namespace std;  int main()  {  setlocale(LC\_CTYPE, "Russian");  int n, x, counter\_max = 0,counter\_min=0,min=0,max=0;  const int max\_n = 99;//ограничиваем максимальный размер  cout << "Введите размер массива(нечетное число =< 99): ";  cin >> n;  if (max\_n < n) {//проверка на введеный пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  if (n%2==0) {//проверка на введеный пользователем размер  cout << "Размер массива неверного формата!";  return 0;  }  int array[max\_n];//объявляем массив  srand((unsigned)time(NULL));//с помощью функции srand(seed of random) генерируем новые числа при каждом запуске,функция time(NULL),возвращает текущее время в секундах  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы массива, наш произвольный размер достигается за счёт того, что мы ведем наш цикл ввода и остальные лишь до указанного размера, тем самым игнорируя остальные элементы. Не задавая им значения, в функции int main они примут значение 0  {  array[i] = rand() % 100;//генерируем по одному элементу массива за шаг  for (int j = 0; j < i; j++) {//проверяем есть ли уже в массиве то число,которое сгенерировали  if (array[i] == array[j] and i!=j) {  i--;//если такое есть,то делаем шаг назад и генерируем новое  }  }  }  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы массива, наш произвольный размер достигается за счёт того, что мы ведем наш цикл ввода и остальные лишь до указанного размера, тем самым игнорируя остальные элементы. Не задавая им значения, в функции int main они примут значение 0  {  cout << array[i] << " ";//выводим элементы массива для возможности их анализа  }  for (int i = 0; i < n; i++)//исследуем созданный массив  {  for (int j = 0; j < n; j++) {//сравниваем каждый элемент массива друг с другом  if (array[i] > array[j]) { counter\_min++; }//находим кол-во элементов,которых элемент с индексом i меньше  if (array[i] < array[j]) { counter\_max++; }//находим кол-во элементов,которых элемент с индексом i больше  }  if (counter\_max == counter\_min ) { cout <<endl<<endl <<"Искомое число:" << array[i]<<endl; return 0; }//если количество таких элементов равно,то это и является искомым числом  counter\_max = 0;//обнуляем счетчики для проверки следующего элемента  counter\_min = 0;  }  } |
| **Результат программы** |
|  |

4.В массиве M, размером k, много совпадающих элементов. Найти количество различных элементов в нем (не упорядочивая массив).

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>//директива, необходимая для функции time  using namespace std;  int main()  {  setlocale(LC\_CTYPE, "Russian");  int n, x, counter = 0,counter\_min=0;  bool check=true;//объявляем переменную буллевого типа для проверки на уникальность  const int max\_n = 100;//ограничиваем максимальный размер  cout << "Введите размер массива( =< 100): ";  cin >> n;  if (max\_n < n) {//проверка на введеный пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  int array[max\_n],uniq[max\_n];//т.к. в условиях нет ограничения на использования доп. массивов,то воспользуемся им для хранения уникальных элементов  srand((unsigned)time(NULL));  for (int i = 0; i < n; i++)  {  array[i] = rand() % 10;//я выбрал диапазон чисел [0;10],т.к. в нем будет немало идентичных элементов  cout << array[i] << " ";//выводим элементы массива для возможности их анализа  for (int j = 0; j < i; j++) {//сравниваем текущий элемент массива с известными уникальными элементами  if (array[i] == uniq[j]) { check = false; }//если текущий элемент уже встречался,то наш элемент не прошел проверку на уникальность  }  if (check==true) {//если элемент проходит проверку,то мы записываем его в наш банк уникальных элементов  uniq[counter] = array[i];  counter++;//чтобы уникальные элементы не записывались в одну ячейку,мы используем переменную counter для продвижения по уникальному массиву  }  check = true;//обновляем переменную проверки для следующего элемента  }  cout << endl<<"Уникальный массив:";  for (int i = 0; i < counter; i++)//выводим уникальный массив для возможности анализа этого массива и работоспособности программы  {    cout << uniq[i] << " ";  }  cout << endl << "Кол-во уникальных элеметов:" << counter; //выводим ответ  } |
| **Результат программы** |
|  |