Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа №10

**Одномерные массивы**

Выполнил: Мышковец Артём Витальевич 10ПИ

Минск 2024

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Выполнить программу, записанную в правой части. Опробовать второй вариант генерации чисел, записанный в комментарии.  Добавить в программу операторы вычисления суммы элементов массива А.  Произвести отладку. | #include <iostream>  #include <ctime>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  const int N = 100;  int i, sz, A[N], rmn = 0, rmx = 99, sum = 0;  cout << "Введите размер массива ";  cin >> sz;  cout << "Массив А:" << endl;  srand((unsigned)time(NULL));  for (i = 0;i < sz;i++) {  A[i] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  cout << A[i] << endl;  sum += A[i];  }  cout << "Сумма: " << sum;  } |
| 2. Выполнить программу, записанную в правой части. Записать ее условие.  Добавить в программу операторы вычисления среднего значения элементов исходного массива.  Произвести отладку. | **(при вводе 4 происходит выход за пределы массива)**  #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  int i, k, size = 4;  float massivA[] = { 5, -4, 17.1, 9, 1 };  int sum = 0;  cout << "Введите номер элемента (от 0 до 4)";  cin >> k;    for (i = k; i <= size; i++)  massivA[i] = massivA[i + 1];  size--;  for (i = 0; i <= size; i++) {  cout << massivA[i] << endl;  sum += massivA[i];  }  cout << "СА= " << sum / size;  }  **Условие: Удалить из данного массива элемент с индексом введеным пользователем** |
| 3. Выполнить программу, приведенную в правой части.  Внести изменения с тем, чтобы вычислялся минимальный элемент массива.  Произвести отладку. | #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  const int maxSize = 30;  int n, i, kmax = 0, minI = 0;  int a[maxSize];  cout << "Введите размер массива:";  cin >> n;  if (n > 30)  return 0;  srand((unsigned)time(NULL));  for (i = 0;i < n;i++) {  a[i] = rand() % 30;  cout << a[i] << " ";  }  cout << endl;  for (i = 1;i < n;i++) {  if (a[i] > a[kmax]) {  kmax = i;  }  }  cout << "Max: " << a[kmax] << endl;  for (i = 1;i != n;++i) {  if (a[i] < a[minI]) {  minI = i;  }  }  cout << "Min: " << a[minI];  } |
| 4. Выполнить прокрутку программы, приведенной в правой части. Записать условие.  Опробовать программу для массивов A и B разного размера. | Прокрутка:  Ввод элементов массива masA:  Пользователь вводит, например: 1, 3, 5, 7, 9  Ввод элементов массива masB:  Пользователь вводит, например: 2, 4, 6, 8, 10  Начало слияния массивов:  k = 0, j = 0, i = 0  Цикл do-while:  Первая итерация:  Сравниваем masA[0] = 1 и masB[0] = 2  1 <= 2, поэтому masC[0] = 1  k = 1, i = 1  Вторая итерация:  Сравниваем masA[1] = 3 и masB[0] = 2  3 > 2, поэтому masC[1] = 2  j = 1, i = 2  Третья итерация:  Сравниваем masA[1] = 3 и masB[1] = 4  3 <= 4, поэтому masC[2] = 3  k = 2, i = 3  Четвертая итерация:  Сравниваем masA[2] = 5 и masB[1] = 4  5 > 4, поэтому masC[3] = 4  j = 2, i = 4  Продолжаем аналогично до i < 10.  Заполнение оставшихся элементов:  Если k == size, копируем оставшиеся из masB.  Если j == size, копируем оставшиеся из masA.  Вывод массива masC:  Результат: 1 2 3 4 5 6 7 8 9 10  #include <stdio.h>  void main()  {  const int size = 5;  int masA[size], masB[size], masC[size \* 2];  int k = 0, j = 0, i = 0;  printf("A:\n");  for (int n = 0; n < size; n++)  scanf\_s("%d", &masA[n]);  printf("B:\n");  for (int n = 0; n < size; n++)  scanf\_s("%d", &masB[n]);  do  {  if (masA[k] <= masB[j])  masC[i++] = masA[k++];  else  masC[i++] = masB[j++];  if (k == size)  for (; j < size; j++)  masC[i++] = masB[j];  if (j == size)  for (; k < size; k++)  masC[i++] = masA[k];  } while (i < size \* 2);  printf("\n");  for (i = 0; i < size \* 2; i++)  printf("%d ", masC[i]);  printf("\n");  } |
| 5. В соответствии со своим вариантом написать программы по условиям, представленным в таблице ниже. Использовать одномерный массив целых случайных чисел (диапазон от 0 до 99). Размер массива ввести с клавиатуры.  Представить результаты в окне Отладчика.  **Вариант 13**  1. В массиве А каждый элемент, кроме первого, заменить суммой всех предыдущих | #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  int arr[100];  int n;  cout << "Введите размер массива:";  cin >> n;  if (n > 100)  return 0;  srand((unsigned)time(NULL));  for (int i = 0;i < n;i++) {  arr[i] = rand() % 99;  }  int sum = arr[0];  for (int i = 1;i != n;++i) {  sum += arr[i];  arr[i] = sum;  }  for (int i = 0;i != n;++i) {  cout << arr[i] << endl;  }  } |
| 5.2. Задан массив из k символов латинского алфавита. Вывести на экран все символы, которые входят в этот массив по одному разу. | #include <iostream>  using namespace std;  bool isThere(char arr[], char num) {  int size = sizeof(arr) / sizeof(arr[0]);  for (int i = 0; i != size;++i) {  if (arr[i] == num) {  return true;  }  }  return false;  }  int main() {  setlocale(LC\_ALL, "RU");  char arr[100];  char orArr[100];  int size;    cout << "Введите размер: ";  cin >> size;  cout << "Введите символы:" << endl;  for (int i = 0;i != size;++i) {  cin >> arr[i];  }  orArr[0] = arr[0];  int counter = 1;  cout << arr[0];  for (int i = 1;i != size;++i) {  if (!(isThere(orArr, arr[i]))) {  cout << arr[i];  orArr[counter] = arr[i];  counter++;  }  }  } |

**Дополнительные задания**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Имеются результаты n ежедневных измерений количества выпавших осадков. За какую из недель (отрезок времени длиной 7 дней), считая с начала периода измерений, выпало наибольшее количество осадков? | //task 1  #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  const int n = 211; //amout of days  int originData[n];  int shortData[(n + 6) / 7];  //filling massive  for (int i = 0;i != n;++i) {  originData[i] = rand()%100;  }  //segmentation original data and get a sum  int segmentIndex = 0;  for (int i = 0;i < n;i+=7) {  int sum = 0;  for (int j = 0; j != 7 && i + j != n;++j) {  sum += originData[i + j];  }  shortData[segmentIndex++] = sum;  }    //get a maximum index of shortdata  int maxIndex = 0;  for (int i = 1;i != segmentIndex;++i) {  if (shortData[i] > shortData[maxIndex]) {  maxIndex = i;  }  }    cout << "Наибольшее количество осадков выпало на " << maxIndex << " неделе.";  } |
| 3. Найти в массиве наибольшее число подряд идущих одинаковых элементов. | //task 3  #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  const int n = 10;  int data[n];  //filling massive  for (int i = 0;i != n;++i) {  cin >> data[i];  }  //counting same elements  int counter = 1;  int maxCount = 0;  for (int i = 1;i != n;++i) {  if (data[i] == data[i - 1]) {  counter++;  if (counter > maxCount) {  maxCount = counter;  }  }  else {  counter = 1;  }  }  cout << "Наибольшее количество подряд идущих одинаковых элементов: " << maxCount;  } |
| 4. В массиве M, размером k, много совпадающих элементов. Найти количество различных элементов в нем (не упорядочивая массив). | //task 4  #include <iostream>  #include <vector>  using namespace std;  bool isThere(vector<int> arr, int num) {  for (int i = 0; i != arr.size();++i) {  if (arr[i] == num) {  return true;  }  }  return false;  }  int main() {  setlocale(LC\_ALL, "RU");  const int k = 10;  int M[k];  vector<int> data;  for (int i = 0;i != k;++i) {  cin >> M[i];  }    int counter = 1;  data.push\_back(M[0]);  for (int i = 1;i != k;++i) {  if (!(isThere(data, M[i]))) {  counter++;  data.push\_back(M[i]);  }  }  cout << counter;  } |