Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 17

По дисциплине «Основы алгоритмизации и программирования»

На тему «Указатели и ссылки при работе с функциями»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условия задач** |
| --- | --- |
| **9** | 1. Создать одномерный массив, содержащий 15 элементов, наполнить его случайными значениями в интервале от 1 до 200. Определить произведение всех элементов массива, значения которых меньше 50.  2. Дана целочисленная матрица **A** размерности **n** x **m** и вектор **B** размерности **m** x **1**. Если все числа отрицательны, то определить произведение этих массивов. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>  using namespace std;  //с помощью указателя передаем значения динамического массива в функцию  int PROD\_50(int\* array, int size) {//функция для нахождения произвидения всех элементов массива меньше 50  int prod = 1;  for (int i = 0; i < size; i++) {  if (array[i] < 50) { prod \*= array[i]; }  }  return prod;  }  //с помощью указателя передаем значения двойного динамического массива в функцию  bool NEG\_MATRX(int\*\* array,int lenght,int width) {//функция для определения есть ли в матрице отрицательные элементы  for (int i = 0; i < width; i++) {  for (int j = 0; j < lenght; j++) {  if (array[i][j] < 0) { return false; }  }  }  return true;  }  bool NEG\_VECT(int\* array, int lenght) {//функция для определения есть ли в массиве отрицательные элементы  for (int j = 0; j < lenght; j++) {  if (array[j] < 0) { return false; }  }  return true;  }  //с помощью указателя передаем значения одиночного и двойного динамического массива в функцию  void MATRX\_PROD(int\*\* array, int\* vector, int lenght, int width) {//функция для вычисления произвидения матроицы и массива,а также его вывода  int\* product = new int[width];  for (int i = 0; i < width; i++) {  product[i] = 0;  for (int j = 0; j < lenght; j++){  product[i] += vector[j] \* array[i][j];  }  }  for (int i = 0; i < width; i++) {  cout << product[i] << " ";  }  delete[] product;  }  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  cout << "Выберите режим работы программы:" << endl << "1-маccивы" << endl << "2-матрицы" << endl;  int k;  cin >> k;  switch (k) {//с помощью switch выбираем режим работы программмы  case 1: {    srand(time(0));  int\* array = new int[15];//создаем массив указателей на числа типа int с помощью функции new в динамической памяти  for (int i = 0; i < 15; i++) {//генерируем числа в массив  array[i] = (rand() % 199)+1;  }  for (int i = 0; i < 15 ; i++) {//выводим массив  cout << array[i] << " ";  }  cout<<endl<<PROD\_50(array, 15);//с помощью функции выводим нужные данные  delete[] array;  break;  }  case 2: {  int lenght, width;  cout << "Enter lenght: ";  cin >> lenght;  cout << "Enter width: ";  cin >> width;  srand(time(0));  int\*\* array = new int\* [width];//создаем массив указателей типа int с помощью функции new в динамической памяти    for (int i = 0; i < width; i++) {  array[i] = new int[lenght];//для каждого массива указателей создаем массив целочисленных данных  }  int\* vector = new int[lenght];//создаем массив указателей на числа типа int с помощью функции new в динамической памяти  for (int i = 0; i < width; i++) {//генерируем числа в матрицу  for (int j = 0; j < lenght; j++) {  array[i][j] = rand() % 100-25;  }  }  cout << "Матрица:" << endl;  for (int i = 0; i < width; i++) {//выводим матрицу  for (int j = 0; j < lenght; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }  for (int i = 0; i < lenght; i++) {//генерируем числа в массив  vector[i] = rand() % 100-25;  }  cout << "Вектор:" << endl;  for (int j = 0; j < lenght; j++) {//выводим массив  cout << vector[j] << endl;  }  //если находим отрицательный элемент,то находим произвидение матрицы и вектора  if (!NEG\_MATRX(array, lenght, width) or !NEG\_VECT(vector, lenght)) { cout << endl << "Произвидение матриц:" << endl; MATRX\_PROD(array, vector, lenght, width); }  else { cout << "Все положительные"; }  for (int i = 0; i < width; i++) {//с помощью функции delete очищаем выделенную динамическую память для предотвращения утечки памяти  delete[] array[i];  }  delete[] array;  delete[] vector;  break;  }  default:cout << "Некорректный ввод";  }  } |
| **Результат программы** |
|  |

Дополнительные варианты

| **№ варианта** | **Условия задач** |
| --- | --- |
| **6** | 1. Найти сумму минимального и максимального элементов одномерного массива.  2. Проверить, все ли строки матрицы содержат хотя бы один нулевой элемент. Если нет, то заменить значения всех отрицательных элементов матрицы на нулевые. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>  using namespace std;  //с помощью указателя передаем значения динамического массива в функцию  int SUM\_MIN\_MAX(int\* array, int size) {//функция для нахождения произвидения всех элементов массива меньше 50  int sum,max\_el=array[0],min\_el=array[0];  for (int i = 0; i < size; i++) {  max\_el = max(max\_el, array[i]);  min\_el = min(min\_el, array[i]);  sum = max\_el + min\_el;  }  return sum;  }  //с помощью указателя передаем значения двойного динамического массива в функцию  bool NUL\_MATRX(int\*\* array, int lenght, int width) {//функция для определения есть ли в матрице отрицательные элементы  bool check=false;  for (int i = 0; i < width; i++) {  for (int j = 0; j < lenght; j++) {  if (array[i][j] == 0) { check = true; }  }  if (!check) { return false; }  check = false;  }  return true;  }  //с помощью указателя передаем значения двойного динамического массива в функцию и возвращаем его в новом виде  int\*\* NUL\_NEG(int\*\* array, int lenght, int width) {//функция для определения есть ли в матрице отрицательные элементы  for (int i = 0; i < width; i++) {  for (int j = 0; j < lenght; j++) {  if (array[i][j] < 0) { array[i][j] = 0; }  }  }  return array;  }  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  cout << "Выберите режим работы программы:" << endl << "1-маccивы" << endl << "2-матрицы" << endl;  int k;  cin >> k;  switch (k) {//с помощью switch выбираем режим работы программмы  case 1: {  int lenght;  cout << "Enter lenght: ";  cin >> lenght;  srand(time(0));  int\* array = new int[lenght];//создаем массив указателей на числа типа int с помощью функции new в динамической памяти  for (int i = 0; i < lenght; i++) {//генерируем числа в массив  array[i] = (rand() % 199) + 1;  }  for (int i = 0; i < lenght; i++) {//выводим массив  cout << array[i] << " ";  }  cout << endl << SUM\_MIN\_MAX(array, lenght);//с помощью функции выводим нужные данные  delete[] array;  break;  }  case 2: {  int lenght, width;  cout << "Enter lenght: ";  cin >> lenght;  cout << "Enter width: ";  cin >> width;  srand(time(0));  int\*\* array = new int\* [width];//создаем массив указателей типа int с помощью функции new в динамической памяти  for (int i = 0; i < width; i++) {  array[i] = new int[lenght];//для каждого массива указателей создаем массив целочисленных данных  }  for (int i = 0; i < width; i++) {//генерируем числа в матрицу  for (int j = 0; j < lenght; j++) {  array[i][j] = rand() % 5 - 3;  }  }  cout << "Матрица:" << endl;  for (int i = 0; i < width; i++) {//выводим матрицу  for (int j = 0; j < lenght; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }    //если не находим нулевой элемент,то меняем матрицу и выводим ее  if (!NUL\_MATRX(array, lenght, width)) {  array = NUL\_NEG(array, lenght, width);  cout << "Новая матрица:" << endl;  for (int i = 0; i < width; i++) {//выводим матрицу  for (int j = 0; j < lenght; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }  }  else { cout << "В каждом ряду есть 0"; }  for (int i = 0; i < width; i++) {//с помощью функции delete очищаем выделенную динамическую память для предотвращения утечки памяти  delete[] array[i];  }  delete[] array;  break;  }  default:cout << "Некорректный ввод";  }  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| **12** | 1. Найти количество отрицательных элементов, стоящих на чётных местах в одномерном массиве.  2. Проверить, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер. Все элементы столбца с таким же номером уменьшить вдвое. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>  using namespace std;  //с помощью указателя передаем значения динамического массива в функцию  int NEG\_EVEN(int\* array, int size) {//функция для нахождения кол-ва отрицательных чисел на четных местах  int counter=0;  for (int i = 1; i < size; i=i+2) {  if (array[i] < 0) { counter++; }  }  return counter;  }  //с помощью указателя передаем значения двойного динамического массива в функцию  int NEG\_ROW(float\*\* array, int lenght, int width) {//функция для определения столбца,в котором есть отрицательный элемент  for (int i = 0; i < width; i++) {  for (int j = 0; j < lenght; j++) {  if (array[i][j] < 0) { return j; }  }  }  return -1;  }  //с помощью указателя передаем значения двойного динамического массива в функцию и возварщаем его в новом виде  float\*\* COL\_DIV(float\*\* array,int row,int lenght){//функция для уменьшения вдвое значения элементов указанного столбца  for (int j = 0; j < lenght; j++) {  array[j][row] /= 2;  }  return array;  }  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  cout << "Выберите режим работы программы:" << endl << "1-маccивы" << endl << "2-матрицы" << endl;  int k;  cin >> k;  switch (k) {//с помощью switch выбираем режим работы программмы  case 1: {  int lenght;  cout << "Enter lenght: ";  cin >> lenght;  srand(time(0));  int\* array = new int[lenght];//создаем массив указателей на числа типа int с помощью функции new в динамической памяти  for (int i = 0; i < lenght; i++) {//генерируем числа в массив  array[i] = rand() % 100-50 ;  }  for (int i = 0; i < lenght; i++) {//выводим массив  cout << array[i] << " ";  }  cout << endl << NEG\_EVEN(array, lenght);//с помощью функции выводим нужные данные  delete[] array;  break;  }  case 2: {  int lenght, width;  cout << "Enter lenght: ";  cin >> lenght;  cout << "Enter width: ";  cin >> width;  srand(time(0));  float\*\* array = new float\* [width];//создаем массив указателей типа float с помощью функции new в динамической памяти  for (int i = 0; i < width; i++) {  array[i] = new float[lenght];//для каждого массива указателей создаем массив данных действительных чисел  }  for (int i = 0; i < width; i++) {//генерируем числа в матрицу  for (int j = 0; j < lenght; j++) {  array[i][j] = rand() % 100 - 15;  }  }  cout << "Матрица:" << endl;  for (int i = 0; i < width; i++) {//выводим матрицу  for (int j = 0; j < lenght; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }  //если находим отрицательный элемент,то меняем матрицу и выводим ее  if (NEG\_ROW(array, lenght, width)>-1) {  cout << endl << "Номер первого столбца с отриц. элементом:" << NEG\_ROW(array, lenght, width)+1 << endl;  array = COL\_DIV(array, NEG\_ROW(array, lenght, width), lenght);  cout << "Новая матрица:" << endl;  for (int i = 0; i < width; i++) {//выводим матрицу  for (int j = 0; j < lenght; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }  }  else { cout << "Отрицательные элементы отсутствуют в матрице"; }  for (int i = 0; i < width; i++) {//с помощью функции delete очищаем выделенную динамическую память для предотвращения утечки памяти  delete[] array[i];  }  delete[] array;  break;  }  default:cout << "Некорректный ввод";  }  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| **4** | 1. Создать одномерный массив, содержащий 15 элементов, наполнить его случайными значениями в интервале от 1 до 200. Определить сумму всех нечетных элементов массива.  2. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>  using namespace std;  //с помощью указателя передаем значения динамического массива в функцию  int SUM\_ODD(int\* array, int size) {//функция для нахождения суммы чисел на нечетных местах  int sum = 0;  for (int i = 0; i < size; i = i + 2) {  sum+=array[i];  }  return sum;  }  //с помощью указателя передаем значения двойного динамического массива в функцию  int NEG\_ROW(float\*\* array, int lenght, int width) {//функция для определения строки,в котором есть положительный элемент  for (int i = 0; i < width; i++) {  for (int j = 0; j < lenght; j++) {  if (array[i][j] > 0) { return i; }  }  }  return -1;  }  //с помощью указателя передаем значения двойного динамического массива в функцию и возварщаем его в новом виде  float\*\* ROW\_NEG(float\*\* array, int col, int lenght) {//функция для изменения знака предшествующей строки,если такое возможно  if(col>=0) {  for (int j = 0; j < lenght; j++) {  array[col][j] \*= -1;  }  }  return array;  }  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  cout << "Выберите режим работы программы:" << endl << "1-маccивы" << endl << "2-матрицы" << endl;  int k;  cin >> k;  switch (k) {//с помощью switch выбираем режим работы программмы  case 1: {    srand(time(0));  int\* array = new int[15];//создаем массив указателей на числа типа int с помощью функции new в динамической памяти  for (int i = 0; i < 15; i++) {//генерируем числа в массив  array[i] = (rand() % 199)+1;  }  for (int i = 0; i < 15 ; i++) {//выводим массив  cout << array[i] << " ";  }  cout<<endl<<SUM\_ODD(array, 15);//с помощью функции выводим нужные данные  delete[] array;  break;  }  case 2: {  int lenght, width;  cout << "Enter lenght: ";  cin >> lenght;  cout << "Enter width: ";  cin >> width;  srand(time(0));  float\*\* array = new float\* [width];//создаем массив указателей типа float с помощью функции new в динамической памяти  for (int i = 0; i < width; i++) {  array[i] = new float[lenght];//для каждого массива указателей создаем массив данных действительных чисел  }  for (int i = 0; i < width; i++) {//генерируем числа в матрицу  for (int j = 0; j < lenght; j++) {  array[i][j] = rand() % 100 - 85;  }  }  cout << "Матрица:" << endl;  for (int i = 0; i < width; i++) {//выводим матрицу  for (int j = 0; j < lenght; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }  //если находим положительный элемент,то меняем матрицу и выводим ее  if (NEG\_ROW(array, lenght, width) > -1) {  cout << endl << "Номер первой строки с полож. элементом:" << NEG\_ROW(array, lenght, width) + 1 << endl;  array = ROW\_NEG(array, NEG\_ROW(array, lenght, width)-1, lenght);  cout << "Новая матрица:" << endl;  for (int i = 0; i < width; i++) {//выводим матрицу  for (int j = 0; j < lenght; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }  }  else { cout << "Положительные элементы отсутствуют в матрице"; }  for (int i = 0; i < width; i++) {//с помощью функции delete очищаем выделенную динамическую память для предотвращения утечки памяти  delete[] array[i];  }  delete[] array;  break;  }  default:cout << "Некорректный ввод";  }  } |
| **Результат программы** |
|  |