**Лабораторная работа № 17. Указатели и ссылки при работе с функциями**

Функция пользователя может принимать *параметры* в виде указателей и ссылок, а также возвращать указатели и ссылки в качестве *результата*.

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Выполнив программу, записанную в правой части, изучить один из способов передачи *динамической* *матрицы* в функцию пользователя.  Записать условие задачи. | |  |  | | --- | --- | | **#include <iostream>**  **using namespace std;**  **int sum(int \*\*, int, int);**    **int main()**  **{ int i, j;**  **int \*\*matr = new int\*[5];**  **for (i = 0; i < 5; i++)**  **matr[i] = new int[4];**  **for(i = 0; i < 5; i++)**  **{ for(j = 0; j < 4; j++)**  **{ matr[i][j] = i + j;**  **cout<< matr[i][j] <<' ';**  **}**  **cout <<endl;**  **}**  **cout<<"S="<< sum(matr, 5, 4) << endl;**  **for(int i = 0; i < 5; i++)**  **delete matr[i];**  **delete [] matr;**  **}** | **int sum(int \*\*x, int n, int m)**  **{ int res = 0;**  **for(int i = 0; i < n; i++)**  **for(int j = 0; j < m; j++)**  **res += x[i][j];**  **return res;**  **}**    **Вычисляет сумму элементов матрицы** | |  |  | |
| 2. Выполнив программу, записанную в правой части, изучить один из способов передачи *одномерного* *массива* в функцию, а также использование *указателя как результата* выполнения функции. | *Указатель-результат* функции может указывать и на отдельную переменную, и на массив.  **#include <iostream>**  **int \*pfmin(int \*p, int n);**  **void main()**  **{ int masB[5] = { 4, 8, 2, 6, 4 };**  **(\*pfmin(masB, 5))++;**  **for (int i = 0; i < 5; i++)**  **std::cout<<masB[i]<<' ';**  **}**  **int \*pfmin(int \*p, int n)**  **{ int \*pmin;**  **for (pmin = p; n > 0; p++, n--)**  **if (\*p < \*pmin)**  **pmin = p;**  **return pmin;**  **}**  Пример. Пусть имеется массив **В**. С использованием функции пользователя определить минимальный элемент массива в подпрограмме и в основной программе увеличить его значение на 1. |
| 3. Изучить использование *ссылки* как *результата* работы функции, выполнив программу, записанную в правой части.  Опробовать работу программы с разными значениями массива **А**. | Пример. Массив **А** содержит набор значений. Необходимо в подпрограмме определить минимальный элемент и изменить его на другое значение в основной программе.  **#include <iostream>**  **using namespace std;**  **double &dmin(double A[], int size);**  **void main()**  **{**  **double s;**  **const int size = 5;**  **double A[] = { 5, 4.1, 3, 0.2, 11 };**  **s = dmin(A, size);**  **cout << s<< endl;**  **for (int i = 0; i < size; i++)**  **cout << " " << A[i];**  **cout << endl;**  **dmin(A, size) = 1.0; // изменение минимума на значение 1.0**  **for (int i = 0; i < size; i++)**  **cout << " " << A[i];**  **}**  **double &dmin(double A[], int size)**  **{**  **int i, j = 0;**  **for (i = 1; i < size; i++)**  **if (A[j] > A[i])**  **j = i;**  **return A[j];**  **}**    Ссылка − это псевдоним, и значит объект, на который она ссылается, существует после выполнения функции пользователя.  Если результат функции пользователя представлен в виде ссылки, то можно использовать вызов функции в левой части оператора присваивания. |
| 4. Ознакомиться с использованием функций с результатом *логического* типа, опробовав работу программы в правой части. | Пример. Имеется массив целых чисел **А**. Определить, содержит ли он число, которое вводится с клавиатуры.  **#include <iostream>**  **using namespace std;**  **bool is\_elem(int \*pA, int n, int iV);**  **void main()**  **{ setlocale(LC\_CTYPE, "Rus");**  **bool t; int k;**  **int A[] = { 5, 4, 3, 2, 11 };**  **cout << "Введите число ";**  **cin >> k;**  **t = is\_elem(A, sizeof(A)/sizeof(int), k);**  **if (t == true)**  **cout << "Число есть в массиве ";**  **else**  **cout << "Числа нет в массиве ";**  **}**  **bool is\_elem(int \*pA, int n, int iV)**  **{ bool bf = false;**  **for (int i = 0; i < n; i++)**  **if (pA[i] == iV)**  **{ bf = true;**  **break;**  **}**  **return bf;**  **}**  Результат, возвращаемый функцией **bool is\_elem ()**, − это логическая переменная, которая может принимать два значения − **true** (если искомый элемент есть в массиве) и **false** (если элемент отсутствует). |

5. В соответствии со своим вариантом написать главную функцию, в которой имеются вызовы ***функций пользователя***, реализующих задачи из таблицы ниже. Ввод исходных данных и вывод результатов осуществить в главной функции, при этом использовать***динамические* *массивы***.

Для передачи параметров в функции пользователя и возвращения результатов применить ***указатели*** и ***ссылки***.

| **№ варианта** | **Условие задачи** |
| --- | --- |
| 3 | 1. В массиве из целых чисел определить сумму элементов, расположенных между первым и последним нулевыми элементами   #include <iostream>  using namespace std;  int\* checkMassive(int n, int \*arr) {  int firstnull = 0, secondnull = 0; int sum = 0;  for (int i = 0; i < n; i++) {  if (arr[i] == 0) {  firstnull = i;  break;  }  }  for (int i = firstnull + 1; i < n; i++) {  if (arr[i] == 0) {  secondnull = i;  break;  }  }  for (int i = firstnull; i < secondnull; i++) {  sum = sum + arr[i];  }  cout << sum;  return 0;  }  int main() {  setlocale(LC\_ALL, "RU");  int n;  cout << "Введите размер массива: "; cin >> n;  int\* arr = new int[n];  cout << "Введите элементы массива: ";  for (int i = 0; i < n; i++) {  cin >> arr[i];  }  cout << endl << "Сумма элементов между нулевыми элементами = ";  \*checkMassive(n, arr);  }     1. Если есть в матрице строка, все элементы которой положительны, то найти сумму этих элементов. Уменьшить все элементы матрицы на эту сумму.   #include <iostream>  using namespace std;  void checkMatrix(int\*\* arr,int n) {  int sum = 0;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if (arr[i][j] > 0) {  sum += arr[i][j];  }  }  break;  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  arr[i][j] = arr[i][j] + sum;  }  }  }  int main() {  setlocale(LC\_ALL, "RU");  int n;  cout << "Введите n: "; cin >> n;  int\*\* arr = new int\* [n];    for (int i = 0; i < n; i++) {  arr[i] = new int[n];  }  cout << "Введите элементы матрицы: " << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cin >> arr[i][j];  }  }  checkMatrix(arr, n);  cout << "Измененная матрица: " << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << arr[i][j] << " ";  }  cout << endl;  }  } |
| 8 | 1. В массиве вещественных чисел заменить отрицательные элементы их квадратами. Найти в полученном массиве индекс элемента, для которого сумма элементов, стоящих до него, наименее отличается от суммы элементов, стоящих после него.   #include <iostream>  using namespace std;  int\* checkNeg(int n, int\* arr) {    for (int i = 0; i < n; i++) {  if (arr[i] < 0) {  arr[i] = arr[i] \* arr[i];  }  }  return 0;  }  int\* checkElem(int n, int\*arr) {  int index = 0;  for (int i = 0; i < n; i++) {  int elemdo = 0; int elempo = 0;  for (int j = 0; j < i; j++) {  elemdo = elemdo+ arr[j];  }  for (int j = i+1; j < n; j++) {  elempo = elempo + arr[j];  }  if (elemdo==elempo ) {  index = i;  }  }  cout <<"Равные значения до и после элемента: " << index << endl;  return 0;  }  int main() {  setlocale(LC\_ALL, "RU");  int n;  cout << "Введите раззмер массива: "; cin >> n;  int\* arr = new int[n];  for (int i = 0; i < n; i++) {  cin >> arr[i];  }  cout << "Измененный массив: ";  \*checkNeg(n, arr);  for (int i = 0; i < n; i++) {  cout << arr[i] << " ";  }  cout << endl;  \*checkElem(n, arr);  }     1. Проверить, есть ли в матрице хотя бы один столбец, содержащий отрицательный элемент, и найти его номер. Уменьшить элементы найденного столбца вдвое.   #include <iostream>  using namespace std;  void processColumns(int\*\* arr, int n) {  bool found = false;  for (int col = 0; col < n; col++) {  for (int i = 0; i < n; i++) {  if (arr[i][col] < 0) {  found = true;  for (int j = 0; j < n; j++) {  arr[j][col] /= 2;  }  cout << "Столбец " << col + 1 << " содержит отрицательный элемент." << endl;  break;  }  }  if (found) {  break;  }  }  if (!found) {  cout << "В матрице нет столбцов с отрицательными элементами." << endl;  }  }  int main() {  setlocale(LC\_ALL, "RU");  int n;  cout << "Введите размер матрицы: ";  cin >> n;  int\*\* arr = new int\* [n];  for (int i = 0; i < n; i++) {  arr[i] = new int[n];  }  cout << "Введите элементы матрицы:" << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cin >> arr[i][j];  }  }  processColumns(arr, n);  cout << "Изменённая матрица:" << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << arr[i][j] << " ";  }  cout << endl;  }  // Освобождение памяти  for (int i = 0; i < n; i++) {  delete[] arr[i];  }  delete[] arr;  return 0;  } |
| 11 | 1. Изменить одномерный массив, вычеркнув из него нечетные элементы.   #include <iostream>  using namespace std;  int\* checkMassive(int n, int\* arr) {  for (int i = 0; i < n; i++) {  if (arr[i] % 2 == 0) {  cout << arr[i] << " ";  }  }  return 0;  }  int main() {  setlocale(LC\_ALL, "RU");  int n;  cout << "Введите раззмер массива: "; cin >> n;  int\* arr = new int[n];  for (int i = 0; i < n; i++) {  cin >> arr[i];  }  cout << "Массив имеющий только четные элементы: ";  \*checkMassive(n, arr);  }     1. Дана целочисленная квадратная матрица. Если она не содержат отрицательных элементов, то определить сумму элементов в тех строках, где отрицательные элементы отсутствуют.   #include <iostream>  using namespace std;  void checkMatrix(int\*\* arr, int n) {  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if (arr[i][j] < 0) {  cout << "Матрица имеет отрицательные элементы" << endl;  return;  }  }  }  cout << "Все элементы матрицы положительны" << endl;  }  void checkRow(int\*\* arr, int n) {  for (int i = 0; i < n; i++) {  int sum = 0;  for (int j = 0; j < n; j++) {  if (arr[i][j] > 0) {  sum += arr[i][j];  }  }  cout << "Сумма " << i + 1 << " строки = " << sum << endl;    }  }  int main() {  setlocale(LC\_ALL, "RU");  int n; int sum = 0;  cout << "Введите раззмер матрицы: "; cin >> n;  int\*\* arr = new int\*[n];  for (int i = 0; i < n; i++) {  arr[i] = new int[n];  }  cout << "Введите элементы матрицы: " << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cin >> arr[i][j];  }  cout << endl;  }  checkMatrix(arr, n);  checkRow(arr, n);  } |
| 14 | 1. В одномерном массиве найти минимальный и максимальный элементы. Вычислить их разность.   #include <iostream>  using namespace std;  int\* checkMassive(int n, int\* arr) {  int maxel = 0; int minel = arr[0];  for (int i = 0; i < n; i++) {  if (arr[i] > maxel) {  maxel = arr[i];  }  if (minel > arr[i]) {  minel = arr[i];  }  }  cout << minel - maxel << endl;  cout << "Минимальный элемент: " << minel << endl;  cout << "Максимальный элемент: " << maxel << endl;  return 0;  }  int main() {  setlocale(LC\_ALL, "RU");  int n;  cout << "Введите раззмер массива: "; cin >> n;  int\* arr = new int[n];  for (int i = 0; i < n; i++) {  cin >> arr[i];  }  cout << "Разница максимального элемента: ";  \*checkMassive(n, arr);  }     1. Если в матрице имеется столбец, все элементы которого положительны, то знаки элементов предыдущего столбца изменить на противоположные.   #include <iostream>  using namespace std;  void checkMatrix(int\*\* arr, int n) {  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if (arr[i][j] < 0) {  cout << "Матрица имеет отрицательные элементы" << endl;  return;  }  }  }  cout << "Все элементы матрицы положительны" << endl;  }  void checkColumn(int\*\* arr, int n, int col) {  for (int i = 0; i < n; i++) {  arr[i][col] = -arr[i][col];  }  }  int main() {  setlocale(LC\_ALL, "RU");  int n; int sum = 0;  cout << "Введите раззмер матрицы: "; cin >> n;  int\*\* arr = new int\* [n];  for (int i = 0; i < n; i++) {  arr[i] = new int[n];  }  cout << "Введите элементы матрицы: " << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cin >> arr[i][j];  }  cout << endl;  }  checkMatrix(arr, n);  for (int col = 1; col < n; col++) {  checkColumn(arr, n, col - 1);  }  cout << "Измененная матрица:" << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << arr[i][j] << " ";  }  cout << endl;  }    } |