**Лабораторная работа № 15. Динамическое выделение памяти**

|  |  |  |
| --- | --- | --- |
| Задание | | Решение |
| 1. Изучить способы передачи значений *переменных* в функцию, выполнив программы, записанные в правой части.  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;**  **}** |   Условие: Программа которая создает матрицу и инициализирует матрицу, а затем находит сумму элементов матрицы в отдельной функции. |
| 2. Выполнив программу, записанную в правой части, изучить один из способов передачи *одномерного* *массива* в функцию, а также использование *указателя как результата* выполнения функции. | | **#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;**  **}** |
| 3. Изучить использование *ссылки* как *результата* работы функции, выполнив программу, записанную в правой части.  Опробовать работу программы с разными значениями массива **А**. | | **#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];**  **}** |
| 4. Ознакомиться с использованием функций с результатом *логического* типа, опробовав работу программы в правой части. | | **#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;**  **}** |
| 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  10 | | |
| Задание | Код | |
| 1. Найти номера четных элементов, стоящих на нечетных местах в одномерном массиве.  2. Дана целочисленная прямоугольная матрица. Если максимальное число встречается в матрице более одного раза, то вывести его значение и индексы, а также количество повторений. | case 10: {  int part;  cout << "Введите номер задания:";  cin >> part;  switch (part) {  case 1: {  int n;  cout << "Введите количество элементов массива:";  cin >> n;  int\* a = new int[n];  for (int i = 0; i < n; i++) {  cin >> a[i];  }  cout << "Номера чётных на нечётных местах:";  Z10\_1(a, n);  delete[] a;  return 0;  }  case 2: {  int n, m;  cin >> n >> m;  int\*\* a = new int\* [n];  for (int i = 0; i < n; i++) {  a[i] = new int[m];  for (int j = 0; j < m; j++) {  cin >> a[i][j];  }  }  Z10\_2(a, n, m);  for (int i = 0; i < n; i++) {  delete[] a[i];  }  delete[] a;  return 0;  }  }  }  void Z10\_1(int\* a,int n) {  for (int i = 0; i < n; i++) {  if (a[i] % 2 == 0 && (i + 1) % 2 != 0) {  cout << i+1 << ' ';  }  }  }  void Z10\_2(int\*\* a, int n, int m) {  int maxi = 0, maxj = 0, col = 1;  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (a[i][j] > a[maxi][maxj]) {  maxi = i; maxj = j;  col = 1;  continue;  }  if (a[i][j] == a[maxi][maxj]) {  col++;  }  }  }  cout << "Максимальный элемент:" << a[maxi][maxj] << " i=" << maxi << " j=" << maxj << " встречается " << col << " раз";  } | |

Допы

|  |  |
| --- | --- |
| №Варианта 13 | |
| Задание | Код/результат |
| 1. Дан массив **A** вещественного типа, содержащий 20 положительных и отрицательных элементов. Сформировать массив **B** из положительных элементов массива **A**, имеющих четный индекс. Найти сумму квадратов элементов нового массива.  2. Дана целочисленная квадратная матрица. Если минимальное значение находится на главной диагонали, то вывести его значение и индексы. | case 13: {  int part;  cout << "Введите номер задания:";  cin >> part;  switch (part) {  case 1: {  float a[20];  cout << "Введите элементы массива\n";  for (int i = 0; i < 20; i++) {  cin >> a[i];  }  float sum = \*Z13\_1(a, 20);  cout << "\nСумма квадратов элементов:" << sum;  return 0;  }  case 2: {  int n;  cout << "Введите размерность массива:";  cin >> n;  int\*\* a = new int\* [n];  for (int i = 0; i < n; i++) {  a[i] = new int[n];  for (int j = 0; j < n; j++) {  cin >> a[i][j];  }  }  Z13\_2(a, n);  return 0;  }  }  }  float\* Z13\_1(float\* a, int n) {  float sum = 0;  int m = 0;  for (int i = 0; i < n; i++) {  if (a[i] > 0 && i % 2 == 0) {  m++;  }  }  float\* b = new float[m];  int j = 0;  for (int i = 0; i < n; i++) {  if (a[i] > 0 && i % 2 == 0) {  b[j] = a[i];  sum += b[j] \* b[j];  j++;  }  }  cout << "Новый массив:\n";  for (int i = 0; i < m; i++) {  cout << b[i] << ' ';  }  float\* ptr = &sum;  return ptr;  }  void Z13\_2(int\*\* a, int n) {  int mini = 0, minj = 0;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if (a[i][j]<a[mini][minj]) {  mini = i;  minj = j;  }  }  }  if (mini == minj) {  cout << "Минимальный элемент:" << a[mini][minj] << " i=" << mini << " j=" << minj;  }  return;  } |

|  |  |
| --- | --- |
| №Варианта 7 | |
| Задание | Код/результат |
| 1. Создать одномерный массив, содержащий 15 элементов, наполнить его случайными значениями в интервале от 1 до 200. Определить произведение элементов массива с индексами от 2 до 7.  2. Если есть в матрице столбец, все элементы которого отрицательны, то найти среднее арифметическое этих элементов. Вычесть полученное значение из всех элементов матрицы. | case 7: {  int part;  cout << "Введите номер задания:";  cin >> part;  switch (part) {  case 1: {  int a[15];  for (int i = 0; i < 15; i++) {  a[i] = rand() % 200 + 1;  }  int sum = \*Z7\_1(a);  cout << "Произведение 2-7 элементов:" << sum;  return 0;  }  case 2: {  int n, m;  cin >> n >> m;  int\*\* a = new int\* [n];  for (int i = 0; i < n; i++) {  a[i] = new int[m];  for (int j = 0; j < m; j++) {  cin >> a[i][j];  }  }  Z7\_2(a, n, m);  return 0;  }  }  }  int\* Z7\_1(int\* a) {  int sum = 1;  int\* ptr = &sum;  for (int i = 2; i <= 7; i++) {  sum \*= a[i];  }  return ptr;  }  void Z7\_2(int\*\* a, int n, int m) {  int ans = 0;  for (int j = 0; j < m; j++) {  for (int i = 0; i < n; i++) {  if (a[i][j] > 0) {  ans = 0;  break;  }  ans += a[i][j];  if (i + 1 == n) {  ans = ans / n;  }  }  if (ans != 0) break;  }  if (ans == 0) {  cout << "Нет столбца из отрицательных элементов";  return;  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  a[i][j] -= ans;  cout << a[i][j] << ' ';  }  cout << '\n';  }  return;  } |
| №Варианта 2 | |
| Задание | Код/результат |
| 1. В массиве из целых чисел найти в процентах частоту появления каждого из **k** наиболее часто встретившихся чисел (**k** – натуральное число, не превосходящее числа элементов массива).  2. Дана целочисленная прямоугольная матрица. Если есть в матрице нулевые элементы, то определить количество строк, содержащих их. | case 2: {  int part;  cout << "Введите номер задания:";  cin >> part;  switch (part) {  case 1: {  int n;  cout << "Введите размер массива:";  cin >> n;  int\* a = new int[n];  for (int i = 0; i < n; i++) {  cin >> a[i];  }  int k;  cout << "Введите количество чисел для вывода:";  cin >> k;  Z2\_1(a, n, k);  return 0;  }  case 2: {  int n, m;  cin >> n >> m;  int\*\* a = new int\* [n];  for (int i = 0; i < n; i++) {  a[i] = new int[m];  for (int j = 0; j < m; j++) {  cin >> a[i][j];  }  }  cout << \*Z2\_2(a, n, m) << " строк содержат нули";  return 0;  }  }  }  void Z2\_1(int\* a, int n, int k) {  int\* b[2];  b[0] = new int[k];  b[1] = new int[k];  for (int q = 0; q < k; q++) {  for (int i = 0; i < n; i++) {  int sum = 0;  for (int j = 0; j < n; j++) {  if (a[i] == a[j]) {  sum++;  }  }  bool flag = true;  for (int j = 0; j < q; j++) {  if (a[i] == b[0][j]) {  flag = false;  break;  }  }  if (flag) {  if (i == 0) {  b[0][q] = a[i];  b[1][q] = sum;  }  else {  if (b[1][q] < sum) {  b[0][q] = a[i];  b[1][q] = sum;  }  }  }  }  }  for (int i = 0; i < k; i++) {  cout << b[0][i] << ":" << (b[1][i] \* 1.0) / n \* 100 << "%\n";  }  return;  }  int\* Z2\_2(int\*\* a, int n, int m) {  int sum = 0;  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (a[i][j] == 0) {  sum++;  break;  }  }  }  int\* ptr = &sum;  return ptr;  } |