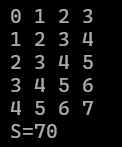
Задание 1

Условие задачи: вывести матрицу каждую строку начиная со значения +1. найти сумму всех элементов матрицы.

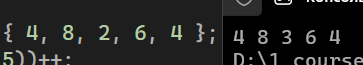
|  |
| --- |
| #include <iostream>  using namespace std;  int sum(int\*\*, int, int);  int main()  {  int i, j;  int\*\* matr = new int\* [5];  for (i = 0; i < 5; i++)  matr[i] = new int[4];  for (i = 0; i < 5; i++)  {  for (j = 0; j < 4; j++)  {  matr[i][j] = i + j;  cout << matr[i][j] << ' ';  }  cout << 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;  } |

Результат программы:



Задание 2

|  |
| --- |
| **#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;**  **}** |



Задание 3

|  |
| --- |
| #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];  }    Изменённый вариант:  #include <iostream>  using namespace std;  double& dmin(double A[], int size);  void main()  {  double s;  const int size = 5;  double A[] = { 0,2,3,4,5 };  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];  } |

Задание 4

|  |
| --- |
| #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;  } |

Вариант 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. |

|  |
| --- |
| #include <iostream>  #include <windows.h>  using namespace std;  int\* task1(int\*, int);  void task2(int\*\*, int, int);  int main() {  SetConsoleCP(1251);  SetConsoleOutputCP(1251);  int choice;  do {  cout << "Какое задание нужно показать" << endl;  cout << "1 - первое задание" << endl;  cout << "2 - второе задание" << endl;  cout << "3 - выход" << endl;  cin >> choice;  switch (choice) {  case 1: {  int n;  cout << "Введите размер одномерного массива: ";  cin >> n;  int\* A = new int[n];  cout << "Введите элементы массива:\n";  for (int i = 0; i < n; i++) {  cin >> A[i];  }  int\* count = task1(A, n);  if (\*count > 0) {  cout << "Количество отрицательных элементов на четных местах: " << \*count << endl;  }  else {  cout << "Таких элементов нет\n";  }  delete count;  delete[] A;  system("pause");  system("cls");  break;  }  case 2: {  int n, m;  cout << "Введите количество строк и столбцов матрицы: ";  cin >> n >> m;  int\*\* matr = new int\* [n];  for (int i = 0; i < n; i++) {  matr[i] = new int[m];  }  cout << "Заполните матрицу:\n";  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cin >> matr[i][j];  }  }  task2(matr, n, m);  for (int i = 0; i < n; i++) {  delete[] matr[i];  }  delete[] matr;  system("pause");  system("cls");  break;  }  case 3:  break;  }  } while (choice != 3);  return 0;  }  int\* task1(int\* A, int n) {  int\* count\_o = new int(0);  for (int i = 0; i < n; i++) {  if (i % 2 == 0 && A[i] < 0) {  (\*count\_o)++;  }  }  return count\_o;  }  void task2(int\*\* matr, int n, int m) {  cout << "Исходная матрица:\n";  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cout << matr[i][j] << " ";  }  cout << endl;  }  bool found = false;  int rowIndex = -1;  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (matr[i][j] < 0) {  rowIndex = i;  found = true;  break;  }  }  if (found) break;  }  if (found) {  cout << "Строка с отрицательным элементом: " << rowIndex + 1 << endl;  cout << "Обновленная матрица (элементы столбца " << rowIndex + 1 << " уменьшены вдвое):\n";  for (int i = 0; i < n; i++) {  matr[i][rowIndex] /= 2;  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cout << matr[i][j] << " ";  }  cout << endl;  }  }  else {  cout << "Нет строк с отрицательными элементами.\n";  }  } |