Задание 1

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  void main()  {  int\* ptr, i, n; // указатель для динамической памяти  printf("Input size of massiv, n < 30 \n");  scanf\_s("%d", &n); // ввод размера массива  // проверка на выделение памяти  if (!(ptr = (int\*)malloc(n \* sizeof(int))))  {  puts("Not enough memory");  return;  }  // ввод элементов массива  for (i = 0; i < n; i++)  {  printf("Input element [%d]\n", i + 1);  scanf\_s("%d", ptr + i); // ввод элемента по адресу  }  // поиск максимального по модулю элемента  int\* max = ptr; // указатель на первый элемент  for (i = 1; i < n; i++)  {  if (abs(\*(ptr + i)) > abs(\*max)) // сравнение по модулю  {  max = ptr + i; // изменяем указатель на новый максимум  }  }  // вывод массива  printf("\nMassiv:\n");  for (i = 0; i < n; i++)  {  printf("%d ", \*(ptr + i));  }  // вывод максимального по модулю элемента  printf("\nMax= %d\n", \*max);  // освобождение динамической памяти  free(ptr);  } |

Задание 2

|  |
| --- |
| #include<iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  char\* pStr;  char c, \* pc;  int j, length, count = 0;  cout << "Введите размер строки ";  cin >> length;  pStr = new char[length + 1];  cout << "Введите строку ";  cin >> pStr;  cout << "Введите символ ";  cin >> c;  pc = &c;  for ( j = 0; j < length; j++)  {  if (pStr[j]==\*pc)  {  count++;  }  }  cout << count << " повторений символа " << c << endl;  delete[] pStr;  } |

Задание 3

|  |
| --- |
| #include <ctime>  #include <iostream>  using namespace std;  void main()  {  int size, mult = 1, sum = 0, \*\* A;  cout << "Enter size ";  cin >> size;  srand(time(0));  A = new int\* [size]; //выделение динам.памяти  for (int i = 0; i < size; i++)  {  A[i] = new int[size];  for (int j = 0; j < size; j++)  A[i][j] = 1 + rand() % 10;  }  for (int\*\* ptr = A; ptr != A + size; ++ptr)  {  for (int\* it = \*ptr; it != \*ptr + size; ++it)  {  cout << "\t" << \*it;  if (\*it % 2 == 0)  {  mult \*= \*it;  sum += \*it;  }  }  cout << endl;  }  cout << "sum: " << sum << endl;  cout << "mult: " << mult << endl;  for (int k = 0; k < size; k++) //освобождение памяти  delete[] A[k];  delete[] A;  } |

Вариант 12

|  |
| --- |
| Задан массив **A** из **n** элементов. Подсчитать, сколько раз встречается в нем максимальное число. |

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  void main()  {  int\* ptr, i, n;  printf("Input size of massiv, n < 30 \n");  scanf\_s("%d", &n);  if (!(ptr = (int\*)malloc(n \* sizeof(int))))  {  puts("Not enough memory");  return;  }  for (i = 0; i < n; i++)  {  printf("Input element [%d]\n", i + 1);  scanf\_s("%d", ptr + i);  }  int count=0;  int\* max = ptr;  for (i = 1; i < n; i++)  {  if ((\*(ptr + i)) > \*max)  {  max = ptr + i;  }  }  for (i = 0; i < n; i++)  {  if (\*(ptr+i)==(\*max))  {  count++;  }  }  printf("\nMassiv:\n");  for (i = 0; i < n; i++)  {  printf("%d ", \*(ptr + i));  }  printf("\nMax= %d\n", \*max);  printf("maximum encountered= %d\n", count);  free(ptr);  } |

|  |
| --- |
| Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент, и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные. |

|  |
| --- |
| #include <iostream>  #include <vector>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  int rows, cols;  cout << "Введите rows: ";  cin >> rows;  cout << "Введите cols: ";  cin >> cols;  int\*\* array = new int\* [rows];  for (int i = 0; i < rows; i++) {  array[i] = new int[cols];  }  cout << "Введите элементы массива: \n";  for (int i = 0; i < rows; i++) {  for (int j = 0; j < cols; j++) {  cout << "Элемент [" << i << "][" << j << "]: ";  cin >> array[i][j];  }  }  cout << "Массив: \n";  for (int i = 0; i < rows; i++) {  for (int j = 0; j < cols; j++) {  cout << array[i][j] << " ";  }  cout << endl;  }  vector<int> num; // тут мы будем храненить номеров строк с положительными элементами  bool pol\_element = false; // есть или нет пол.элемент в строке  for (int i = 0; i < rows; i++) {  bool hasPositive = false; //чтобы не повторялись  for (int j = 0; j < cols; j++) {  if (array[i][j] > 0) {  hasPositive = true;  break;  }  }  if (hasPositive) {  pol\_element = true;  num.push\_back(i);  }  }  cout << "\nНомера строк, содержащих положительный элемент: ";  for (int i = 0; i < num.size(); i++) {  cout << num[i] << " ";  }  cout << endl;  for (int i = 1; i < rows; i++) {  bool hasPositive = false;  for (int j = 0; j < cols; j++) {  if (array[i][j] > 0) {  hasPositive = true;  break;  }  }  if (hasPositive) {  // изменяем знаки элементов предыдущей строки  for (int j = 0; j < cols; j++) {  array[i - 1][j] = -array[i - 1][j];  }  }  }  cout << "\nНовый массив: \n";  for (int i = 0; i < rows; i++) {  for (int j = 0; j < cols; j++) {  cout << array[i][j] << " ";  }  cout << endl;  }  for (int i = 0; i < rows; i++) { //сначала для столбцов,а потом для строк  delete[] array[i];  }  delete[] array;  } |

Доп.задания:

Вариант 11(12-1=11)

|  |
| --- |
| Задан массив **A** из **n** элементов. Проверить, есть ли в нём элементы, равные нулю. Если есть, найти наименьшее **k**, при котором **A[k] = 0**. |

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  void main()  {  int\* A, i, n;  bool en = false;  printf("Input size of massiv, n < 30 \n");  scanf\_s("%d", &n);  if (!(A = (int\*)malloc(n \* sizeof(int))))  {  puts("Not enough memory");  return;  }  for (i = 0; i < n; i++)  {  printf("Input element [%d]\n", i + 1);  scanf\_s("%d", A + i);  }  int cm = -1;  for (int i = 0; i < n; i++)  {  if (\*(A + i)==0)  {  en = true;  cm = i;  break;  }  }  if (en)  {  printf("Naim k= %d\t", cm+1);  }  else  {  printf("\nNet 0");  }  free(A);  } |

|  |
| --- |
| Для заданной матрицы размером 4 на 4 найти такие **k**, при которых **k**-я строка матрицы совпадает с **k**-м столбцом. |

|  |
| --- |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  const int SIZE = 4;  int\*\* array = new int\* [SIZE];  for (int i = 0; i < SIZE; i++) {  array[i] = new int[SIZE];  }  cout << "Введите элементы матрицы 4x4: \n";  for (int i = 0; i < SIZE; i++) {  for (int j = 0; j < SIZE; j++) {  cout << "Элемент [" << i << "][" << j << "]: ";  cin >> array[i][j];  }  }  cout << "Матрица: \n";  for (int i = 0; i < SIZE; i++) {  for (int j = 0; j < SIZE; j++) {  cout << array[i][j] << " ";  }  cout << endl;  }  cout << "\nИндексы k, где k-я строка совпадает с k-м столбцом: ";  bool found = false; // проверки совпадений  for (int k = 0; k < SIZE; k++) {  bool match = true; // проверка совпадения строки и столбца  for (int i = 0; i < SIZE; i++) {  if (array[k][i] != array[i][k]) {  match = false;  break;  }  }  if (match) {  cout << k << " ";  found = true;  }  }  if (!found) {  cout << "Совпадений не найдено.";  }  for (int i = 0; i < SIZE; i++) {  delete[] array[i];  }  delete[] array;  } |

Вариант 3 (12+7)

|  |
| --- |
| В одномерном массиве, состоящем из п вещественных элементов, вычислить произведение отрицательных элементов массива и сумму положительных элементов массива, расположенных до максимального элемента. |

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  void main()  {  float\* A;  int i, n;  bool en = false;  printf("Input size of array, n < 30 \n");  scanf\_s("%d", &n);  if (!(A = (float\*)malloc(n \* sizeof(float))))  {  puts("Not enough memory");  return;  }  for (i = 0; i < n; i++)  {  printf("Input element [%d]\n", i + 1);  scanf\_s("%f", A + i);  }  float max = \*A,sum=0,prz=1;  int count =0;  for (int i = 0; i < n; i++)  {  if (\*(A+i)>max)  {  max = \*(A + i);  count = i;  }  }  for (int i = 0; i < count; i++)  {  if (\*(A + i) < 0)  {  prz \*= (\*(A + i));  }  if (\*(A + i) > 0)  {  sum += (\*(A + i));  }  }  printf("Max.element=%.2f\nCount=%d", max, count+1);  printf("\nProizvedenie=%.2f\nSumma=%.2f", prz, sum);  free(A);  } |

|  |
| --- |
| Дана матрица размером 4x4. Найти сумму наименьших элементов ее нечетных строк и наибольших элементов ее четных строк. |

|  |
| --- |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  const int SIZE = 4;  int\*\* array = new int\* [SIZE];  for (int i = 0; i < SIZE; i++) {  array[i] = new int[SIZE];  }  cout << "Введите элементы матрицы 4x4: \n";  for (int i = 0; i < SIZE; i++) {  for (int j = 0; j < SIZE; j++) {  cout << "Элемент [" << i << "][" << j << "]: ";  cin >> array[i][j];  }  }  cout << "Матрица: \n";  for (int i = 0; i < SIZE; i++) {  for (int j = 0; j < SIZE; j++) {  cout << array[i][j] << " ";  }  cout << endl;  }  int sumNaim = 0, sumNaib = 0;  for (int i = 0; i < SIZE; i++)  {  int naim = 0, naib = 0;  if (i % 2 != 0)  {  naim = array[i][0];  for (int j = 0; j < SIZE; j++)  {  if (array[i][j]<naim)  {  naim = array[i][j];  }  }  sumNaim += naim;  }  else if (i%2==0)  {  naib = array[i][0];  for (int j = 0; j < SIZE; j++)  {  if (array[i][j]>naib)  {  naib = array[i][j];  }  }  sumNaib += naib;  }  }  cout << "\n\nSum naim.nech.=" << sumNaim << "\nSum naib.chet.=" << sumNaib;  for (int i = 0; i < SIZE; i++) {  delete[] array[i];  }  delete[] array;  } |

Вариант 14(12+2)

|  |
| --- |
| В массиве, состоящем из вещественных элементов, вычислить количество элементов массива, больших некоторого заданного числа **С**, и произведение элементов массива, расположенных после максимального по модулю элемента. |

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include<cmath>  void main()  {  float\* A,C;  int i, n;  printf("Input size of array, n < 30 \n");  scanf\_s("%d", &n);  if (!(A = (float\*)malloc(n \* sizeof(float))))  {  puts("Not enough memory");  return;  }  for (i = 0; i < n; i++)  {  printf("Input element [%d]\n", i + 1);  scanf\_s("%f", A + i);  }  printf("Input C: ");  scanf\_s("%f", &C);  float max = fabs(\*A), sum = 0, prz = 1;  int count = 0,count\_C=0;  for (int i = 0; i < n; i++)  {  if (fabs(\*(A + i)) > max)  {  max = fabs(\*(A + i));  count = i;  }  if (\*(A + i)>C)  {  count\_C++;  }  }  for (int i = count + 1; i < n; i++) {  prz \*= A[i];  }  printf("\nMax=%.2f\nProizv=%.2f\nCount > C=%d", max, prz,count\_C);  free(A);  }  **сделать скрины** |

|  |
| --- |
| Проверить, есть ли в матрице хотя бы одна строка, содержащая элемент, равный нулю, и найти ее номер. Уменьшить все элементы матрицы на значение первого элемента найденной строки. |

|  |
| --- |
| #include <iostream>  #include <vector>  using namespace std;  void main() {  setlocale(LC\_CTYPE, "Russian");  int rows, cols;  cout << "Введите количество строк: ";  cin >> rows;  cout << "Введите количество столбцов: ";  cin >> cols;  int\*\* array = new int\* [rows];  for (int i = 0; i < rows; i++) {  array[i] = new int[cols];  }  cout << "Введите элементы массива: \n";  for (int i = 0; i < rows; i++) {  for (int j = 0; j < cols; j++) {  cout << "Элемент [" << i << "][" << j << "]: ";  cin >> array[i][j];  }  }  int rz = -1;  for (int i = 0; i < rows; i++) {  for (int j = 0; j < cols; j++) {  if (array[i][j] == 0) {  rz = i;  break;  }  }  if (rz != -1) {  break;  }  }  if (rz != -1) {  cout << "\nСтрока, содержащая ноль, найдена: " << rz + 1 << endl;  int first\_element = array[rz][0];  for (int i = 0; i < rows; i++) {  for (int j = 0; j < cols; j++) {  array[i][j] -= first\_element;  }  }  cout << "\nИзмененная матрица:\n";  for (int i = 0; i < rows; i++) {  for (int j = 0; j < cols; j++) {  cout << array[i][j] << " ";  }  cout << endl;  }  }  else {  cout << "\nВ матрице нет строки, содержащей ноль.\n";  }  for (int i = 0; i < rows; i++) {  delete[] array[i];  }  delete[] array;  } |