Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 15

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамическое выделение памяти»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условия задач** |
| --- | --- |
| **9** | 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить произведение положительных элементов массива и сумму элементов массива, расположенных до минимального элемента.  2. Найти в матрице первую строку, все элементы которой положительны, и сумму этих элементов. Уменьшить все элементы матрицы на эту сумму. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <stdlib.h>//библиотека для функций работы с динамической памятью  using namespace std;  int main()  {  system("color 70");  int\* ptr, i, n,pos\_counter=1,min\_elind=0,sum=0;  cout<<"Input size of massiv, n<30:";  cin>>n;  if (!(ptr = (int\*)malloc(n \* sizeof(int)))) //выделение памяти с помощью функции malloc и проверка,  { //достаточно ли для нее места  puts("Not enough memory");  return 0;  }  for (i = 0; i < n; i++)//вводим элементы в массив  {  cout<<"Input element int "<< i + 1<<": ";  cin >> \*(ptr + i);  }  cout << endl << "Massiv: " << endl;//вывод массива и поиск минимального элемента  for (i = 0; i < n; i++){  cout << \*(ptr + i) << " ";  if (\*(ptr + min\_elind) != min(\*(ptr + min\_elind), \*(ptr + i))) { min\_elind = i; }  }  cout << endl << "Min element and index: "<<\*(ptr+min\_elind)<<" "<<min\_elind+1 << endl;  for (i = 0; i < min\_elind; i++) {//подсчет произведения положительных элементов и суммы всех элементов до мин элемента  sum += \*(ptr + i);  if (\*(ptr + i) > 0) { pos\_counter \*= \*(ptr + i); }  }  cout << endl << "Product: " << pos\_counter << endl;  cout << "Sum: " << sum<< endl;  free(ptr); //освобождение динамической памяти  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <ctime>  #include <iostream>  using namespace std;  int main()  {  system("color 70");  int size, sum = 0, ind\_pos;  cout << "Enter size ";  cin >> size;  ind\_pos = size+1;  srand(time(0));  int\*\* array = new int\* [size];//создаем массив указателей типа int с помощью функции new в динамической памяти  for (int i = 0; i < size; i++) {  array[i] = new int[size];//для каждого массива указателей создаем массив целочисленных данных  }  for (int i = 0; i < size; i++) {//генерируем числа в матрицу  for (int j = 0; j < size; j++) {  array[i][j] = (rand() % 40) - 10;  }  }  for (int i = 0; i < size; i++) {//выводим матрицу  for (int j = 0; j < size; j++) {  cout<<array[i][j]<<" \t";  }  cout << endl;  }  for (int i = 0; i < size; i++) {  for (int j = 0; j < size; j++) {  if (array[i][j] < 1) { break; }//если встретили неположительное число, то переходим к следующей строке  if (j == size - 1) {//если до шли до последнего элемента, то запоминаем номер строки и заканчиваем анализ  ind\_pos = i;  i = size;  }  }  }  if (ind\_pos > size) { cout << "No positive line found :("; return 0; }    cout << endl << "Positive line:" << endl << endl;//выводим найденную строку и находим сумму ее элементов  for (int j = 0; j < size; j++) {  cout << array[ind\_pos][j] << " \t";  sum += array[ind\_pos][j];  }  cout << endl << "sum:" << sum << endl;  cout << endl << "New massiv:" << endl << endl;//преобразовываем массив и выводим его  for (int i = 0; i < size; i++) {  for (int j = 0; j < size; j++) {  array[i][j] -= sum;  cout << array[i][j] << " \t";  }  cout << endl;  }  for (int i = 0; i < size; i++) {//с помощью функции delete очищаем выделенную динамическую память для устранения утечки памяти  delete[] array[i];  }  delete[] array;  } |
| **Результат программы** |
|  |

Дополнительные варианты

| **№ варианта** | **Условия задач** |
| --- | --- |
| **2** | 1. Задан массив **A** из **n** элементов. Проверить, есть ли в нём отрицательные элементы. Если есть, то найти наибольшее значение **k**, при котором **A[k] < 0**.  2. Дана матрица. Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент оказался в верхнем левом углу. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <stdlib.h>//библиотека для функций работы с динамической памятью  using namespace std;    int main()  {  system("color 70");  int\* ptr, i, n,pos\_counter=1,max\_elind,sum=0;  cout<<"Input size of massiv:";  cin>>n;  if (!(ptr = (int\*)malloc(n \* sizeof(int)))) //выделение памяти с помощью функции malloc и проверка,  { //достаточно ли для нее места  puts("Not enough memory");  return 0;  }  for (i = 0; i < n; i++)//вводим элементы в массив и находим индекс любого отрицательного элемента  {  cout<<"Input element int "<< i + 1<<": ";  cin >> \*(ptr + i);  if (\*(ptr + i) < 0) { max\_elind = i; }  }  cout << endl << "Massiv: " << endl;//вывод массива и поиск максимального элемента среди отрицательных чисел  for (i = 0; i < n; i++){  cout << \*(ptr + i) << " ";  if(\*(ptr+i)<0)  {  if (\*(ptr + max\_elind) != max(\*(ptr + max\_elind), \*(ptr + i))) { max\_elind = i; }  }  }  cout << endl << "Max neg element and index: "<<\*(ptr+max\_elind)<<" "<<max\_elind+1 << endl;//вывод ответа  free(ptr); //освобождение динамической памяти  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <ctime>  #include <iostream>  using namespace std;  int main()  {  system("color 70");  int size, sum = 0, ind\_pos,max\_el=0;  cout << "Enter size ";  cin >> size;  ind\_pos = size+1;  srand(time(0));  int\*\* array = new int\* [size];//создаем массив указателей типа int с помощью функции new в динамической памяти  for (int i = 0; i < size; i++) {  array[i] = new int[size];//для каждого массива указателей создаем массив целочисленных данных  }  for (int i = 0; i < size; i++) {//генерируем числа в матрицу и ищемм максимальный элемент  for (int j = 0; j < size; j++) {  array[i][j] = (rand() % 100);  max\_el = max(max\_el, array[i][j]);  }  }  for (int i = 0; i < size; i++) {//выводим матрицу  for (int j = 0; j < size; j++) {  cout<<array[i][j]<<" \t";  }  cout << endl;  }  cout << endl << "Max element:" << max\_el << endl;  cout << endl << "New massiv:" << endl << endl;//преобразовываем матрицу  for (int i = 0; i < size; i++) {  for (int j = 0; j < size; j++) {  if (array[i][j] == max\_el) { swap(array[i][j], array[0][0]); }  }  }  for (int i = 0; i < size; i++) {//выводим матрицу  for (int j = 0; j < size; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }  for (int i = 0; i < size; i++) {//с помощью функции delete очищаем выделенную динамическую память для устранения утечки памяти  delete[] array[i];  }  delete[] array;  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| **12** | 1. Задан массив **A** из **n** элементов. Подсчитать, сколько раз встречается в нем максимальное число.  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 <stdlib.h>//библиотека для функций работы с динамической памятью  using namespace std;    int main()  {  system("color 70");  int\* ptr, i, n,counter=0,max\_elind=0;  cout<<"Input size of massiv:";  cin>>n;  if (!(ptr = (int\*)malloc(n \* sizeof(int)))) //выделение памяти с помощью функции malloc и проверка,  { //достаточно ли для нее места  puts("Not enough memory");  return 0;  }  for (i = 0; i < n; i++)//вводим элементы в массив и находим индекс любого отрицательного элемента  {  cout<<"Input element int "<< i + 1<<": ";  cin >> \*(ptr + i);  if (\*(ptr + i) > \*(ptr+max\_elind)) { max\_elind = i; }  }  cout << endl << "Massiv: " << endl;//вывод массива и поиск максимального элемента среди отрицательных чисел  for (i = 0; i < n; i++){  cout << \*(ptr + i) << " ";  if (\*(ptr + i) == \*(ptr + max\_elind)) { counter++; }    }  cout << endl << "Povtori max element: "<<counter++;//вывод ответа  free(ptr); //освобождение динамической памяти  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <ctime>  #include <iostream>  using namespace std;  int main()  {  system("color 70");  int size, sum = 0, ind\_pos;  cout << "Enter size ";  cin >> size;  ind\_pos = size + 1;  srand(time(0));  int\*\* array = new int\* [size];//создаем массив указателей типа int с помощью функции new в динамической памяти  for (int i = 0; i < size; i++) {  array[i] = new int[size];//для каждого массива указателей создаем массив целочисленных данных  }  for (int i = 0; i < size; i++) {//генерируем числа в матрицу  for (int j = 0; j < size; j++) {  array[i][j] = (rand() % 100) - 95;  }  }  for (int i = 0; i < size; i++) {//выводим матрицу  for (int j = 0; j < size; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }  for (int i = 0; i < size; i++) {  for (int j = 0; j < size; j++) {  if (array[i][j]>0) {//если встретили положительное число, то запоминаем номер строки и заканчиваем анализ  ind\_pos = i;  i = size;  break;  }  }  }  if (ind\_pos > size) { cout << "No positive line found :("; return 0; }  cout << endl << "Positive line:" << endl << endl;  for (int j = 0; j < size; j++) {  cout<<array[ind\_pos][j]<<" \t";  }  if (ind\_pos != 0)//если положительная строка не первая, то меняем предшествующюю  {  for (int j = 0; j < size; j++) {  array[ind\_pos - 1][j] \*= -1;  }  }  cout << endl << "New massiv:" << endl << endl;//выводим матрицу  for (int i = 0; i < size; i++) {  for (int j = 0; j < size; j++) {  cout << array[i][j] << " \t";  }  cout << endl;  }  for (int i = 0; i < size; i++) {//с помощью функции delete очищаем выделенную динамическую память для устранения утечки памяти  delete[] array[i];  }  delete[] array;  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| **8** | 1. В одномерном массиве, состоящем из **k** целых элементов, вычислить количество положительных элементов массива и сумму элементов массива, расположенных после последнего элемента, равного нулю.  2. Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <stdlib.h>//библиотека для функций работы с динамической памятью  using namespace std;  int main()  {  system("color 70");  int\* ptr, i, n, counter = 0, nul\_ind,sum=0 ;  cout << "Input size of massiv:";  cin >> n;  nul\_ind = n + 1;  if (!(ptr = (int\*)malloc(n \* sizeof(int)))) //выделение памяти с помощью функции malloc и проверка,  { //достаточно ли для нее места  puts("Not enough memory");  return 0;  }  for (i = 0; i < n; i++)//вводим элементы в массив и находим индекс последнего отрицательного элемента  {  cout << "Input element int " << i + 1 << ": ";  cin >> \*(ptr + i);  if (\*(ptr + i) == 0) { nul\_ind = i; }  }  if (nul\_ind > n) { cout << endl << "No 0 elements :( "; return 0; }  cout << endl << "Massiv: " << endl;//поиск кол-ва положительных элементов и сумма элементов после последнего нуля  for (i = nul\_ind; i < n; i++) {  if (\*(ptr + i) > 0) { counter++; }  sum += \*(ptr + i);  }  for (i = 0; i < n; i++) {//вывод массива  cout << \*(ptr + i) << " ";  }  cout << endl << "Number of positive numbers: " << counter << endl;//вывод ответов  cout << "Sum: " << sum << endl;  free(ptr); //освобождение динамической памяти  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <ctime>  #include <iostream>  using namespace std;  int main()  {  system("color 70");  int lenght,width , sum = 0, ind\_pos;  cout << "Enter lenght: ";  cin >> lenght;  cout << "Enter width: ";  cin >> width;  ind\_pos = width + 1;  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() % 10) - 5;  }  }  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++) {  for (int j = 0; j < width; j++) {  if (array[j][i]==0) {//если встретили ноль, то запоминаем номер строки и заканчиваем анализ  ind\_pos = i;  i = lenght;  break;  }  }  }  if (ind\_pos > width) { cout << "No null line found :("; return 0; }  cout << endl << "Null line:" << endl << endl;//выводим искомый столбец  for (int j = 0; j < width; j++) {  cout<<array[j][ind\_pos]<<endl;  }  for (int i = 0; i < width; i++) {//с помощью функции delete очищаем выделенную динамическую память для предотвращения утечки памяти  delete[] array[i];  }  delete[] array;  } |
| **Результат программы** |
|  |