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

|  |  |  |
| --- | --- | --- |
| Задание | | Решение |
| 1. Изучить формирование динамического массива с использованием стандартных функций, выполнив программу на *языке* ***С***, записанную справа.  Написать пояснения к программе.  Добавить операторы вычисления максимального по модулю элемента массива. | | **#include <stdio.h>**  **#include <stdlib.h>**  **void main()**  **{**  **int \*ptr, i, n;**  **printf("Input size of massiv, n<30 \n");**  **scanf("%d", &n);**  **if(!(ptr = (int\*)malloc(n\*sizeof(int)))) //выделение памяти и проверка,**  **{ //достаточно ли для нее места**  **puts("Not enough memory");**  **return;**  **}**  **for (i = 0; i < n; i++)**  **{**  **printf("Input element [%d]\n" , i + 1);**  **scanf("%d", ptr + i);**  **}**  **printf("\nMassiv: \n", i + 1);**  **for (i = 0; i < n; i++)**  **printf("%d ", \*(ptr + i));**  **free(ptr); //освобождение динамической памяти**  **}**    #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)))) //выделение памяти и проверка,  { //достаточно ли для нее места  puts("Not enough memory");  return;  }  int maxi = 0;  for (i = 0; i < n; i++)  {  printf("Input element [%d]\n", i + 1);  scanf\_s("%d", ptr + i);  if (\*(ptr + i) > \*(ptr + maxi)) {  maxi = i;  }  }  printf("\nMassiv: \n", i + 1);  for (i = 0; i < n; i++)  printf("%d ", \*(ptr + i));  printf("\nMax element: %d", \*(ptr + maxi));  free(ptr); //освобождение динамической памяти  } |
| 2. Изучить способы выделения динамической памяти для *одномерного* *массива*, выполнив программу на *языке* ***С++***, записанную в правой части.  Опробовать работу программы с разными значениями вводимых символов. | | #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  char\* pStr;// указатель для адреса массива  char c, \* pc;  int j, length, count = 0;  cout << "Введите размер строки ";  cin >> length;  pStr = new char[length + 1]; // выделение динам. памяти  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; // освобождение динам, памяти  return;  } |
| 3. В программе, записанной справа, демонстрируется использование динамической памяти при работе с *двумерным массивом*.  Выполнить программу с различными размерами исходного массива. | | **#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;**  **}** |
| 4. В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов для условий задач из таблицы. Начальные размерности массивов ввести с клавиатуры. Первое задание выполнить с использованием ***функций*** для выделения динамической памяти на языке С, второе − с использованием операторов **new** и **delete**  на языке **С++**. | | |
| №Варианта  10 | | |
| Задание | Код | |
| 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить сумму положительных элементов массива и произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами. | int n;  cin >> n;  float\* a = (float\*)malloc(n \* sizeof(float));  float sum = 0;  int maxi = 0, mini = 0;  for (int i = 0; i < n; i++) {  cin >> a[i];  if (a[i] > 0) sum += a[i];  if (abs(a[i]) > abs(a[maxi]))  maxi = i;  if (abs(a[i]) < abs(a[mini]))  mini = i;  }  cout << "Сумма положительных:" << sum;  cout << "\nПроизведение между макс и мин:";  if (mini > maxi) {  int temp = maxi;  maxi = mini;  mini = temp;  }  if (abs(maxi - mini) == 1) {  cout << 0;  return 0;  }  int mult = 1;  for (int i = mini + 1; i < maxi; i++) {  mult \*= a[i];  }  cout << mult;  free(a);  return 0; | |
| 2. Найти в матрице первую строку, все элементы которой отрицательны. Увеличить все элементы матрицы на значение первого элемента найденной строки | int n, m;  cin >> m >> n;  int\*\* a = new int\* [n];  for (int i = 0; i < n; i++) {  a[i] = new int[m];  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cin >> a[i][j];  }  }  int i1 = -1;  cout << '\n';  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (a[i][j] > 0) {  break;  }  if (j == m - 1) {  i1 = i;  }  }  if (i1 != -1) break;  }  if (i1 == -1) {  cout << "Нет строки из отрицательных";  return 0;  }  cout << "Изменённый массив:\n";  int temp = a[i1][0];  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  a[i][j] += temp;  cout << a[i][j] << ' ';  }  cout << '\n';  }  for (int i = 0; i < n; i++) {  delete[] a[i];  }  delete[] a;  return 0; | |

Допы

|  |  |
| --- | --- |
| №Варианта 9 | |
| Задание | Код/результат |
| 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить произведение положительных элементов массива и сумму элементов массива, расположенных до минимального элемента. | int n;  cin >> n;  float\* a = (float\*)malloc(n \* sizeof(float));  float mult = 1;  int mini = 0;  for (int i = 0; i < n; i++) {  cin >> a[i];  if (a[i] > 0) {  mult \*= a[i];  }  if (a[i] < a[mini]) {  mini = i;  }  }  float sum = 0;  for (int i = 0; i < mini; i++) {  sum += a[i];  }  cout << "Произведение положительных элементов:" << mult;  cout << "\nСумма до минимального:" << sum;  free(a);  return 0; |
| 2. Найти в матрице первую строку, все элементы которой положительны, и сумму этих элементов. Уменьшить все элементы матрицы на эту сумму. | 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 i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cin >> a[i][j];  }  }  cout << '\n';  int i1 = -1;  int sum = 0;  for (int i = 0; i < n; i++) {  sum = 0;  for (int j = 0; j < m; j++) {  if (a[i][j] < 0) {  break;  }  sum += a[i][j];  if (j == m - 1) {  i1 = i;  }  }  if (i1 != -1) break;  }  if (i1 == -1) {  cout << "Нет строки из положительных";  return 0;  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  a[i][j] -= sum;  cout << a[i][j] << ' ';  }  cout << '\n';  }  for (int i = 0; i < n; i++) {  delete[] a[i];  }  delete[] a;  return 0; |

|  |  |
| --- | --- |
| №Варианта 1 | |
| Задание | Код/результат |
| 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить количество отрицательных элементов массива и сумму модулей элементов, расположенных после минимального по модулю элемента. | int n;  cin >> n;  float\* a = (float\*)malloc(n \* sizeof(float));  int k = 0;  int mini = 0;  for (int i = 0; i < n; i++) {  cin >> a[i];  if (a[i] < 0) {  k++;  }  if (abs(a[i]) < abs(a[mini])) {  mini = i;  }  }  float sum = 0;  for (int i = mini; i < n; i++) {  sum += abs(a[i]);  }  cout << "Количество отрицательных элементов:" << k;  cout << "\nСумма после минимального:" << sum;  free(a);  return 0; |
| 2. Найти в матрице первый столбец, все элементы которого положительны. Знаки элементов предыдущего столбца изменить на противоположные. | 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 i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cin >> a[i][j];  }  }  cout << '\n';  int j1 = -1;  for (int j = 0; j < m; j++) {  for (int i = 0; i < n; i++) {  if (a[i][j] <= 0) {  break;  }  if (i == n - 1) {  j1 = j;  }  }  if (j1 != -1) break;  }  if (j1 == -1) {  cout << "Нет столбца из положительных";  return 0;  }  if (j1 == 0) {  cout << "Предыдущего столбца нет";  return 0;  }  for (int i = 0; i < n; i++) {  a[i][j1 - 1] \*= -1;  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cout << a[i][j] << ' ';  }  cout << '\n';  }  for (int i = 0; i < n; i++) {  delete[] a[i];  }  delete[] a;  return 0; |
| №Варианта 12 | |
| Задание | Код/результат |
| 1. Задан массив **A** из **n** элементов. Подсчитать, сколько раз встречается в нем максимальное число. | int n;  cin >> n;  float\* a = (float\*)malloc(n \* sizeof(float));  int k = 0;  int maxi = 0;  for (int i = 0; i < n; i++) {  cin >> a[i];  if (a[i] > a[maxi]) {  maxi = i;  k = 0;  }  if (a[i] == a[maxi]) {  k++;  }  }  cout << "Максимальный элемент:" << a[maxi];  cout << "\nКоличество максимальных элементов:" << k;  free(a);  return 0; |
| 2. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент, и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные. | 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 i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cin >> a[i][j];  }  }  cout << '\n';  int i1 = -1;  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (a[i][j] > 0) {  if (i != 0) {  i1 = i;  break;  }  else {  i1 = -2;  }  }  }  if (i1 != -1) break;  }  if (i1 == -1) {  cout << "Нет строки c положительным";  return 0;  }  if (i1 == -2) {  cout << "Нет предыдущей строки для найденной";  return 0;  }  for (int j = 0; j < m; j++) {  a[i1][j] \*= -1;  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cout << a[i][j] << ' ';  }  cout << '\n';  }  for (int i = 0; i < n; i++) {  delete[] a[i];  }  delete[] a; |