**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. | | |  |  |  | | --- | --- | --- | | Передача параметра  по значению | Передача параметра  по указателю | Передача параметра  по ссылке | | **int square(int);**  **void main()**  **{ for(int x=1; x<=10; x++)**  **cout<<square(x)<<' ';**  **}**  **int square(int y)**  **{ return y \* y; }** | **int square(int \*);**  **void main()**  **{ for(int x=1; x<=10; x++)**  **cout<<square(&x)<<' ';**  **}**  **int square(int\* y)**  **{ return \*y \* \*y; }** | **int square(int &);**  **void main()**  **{ for(int x=1; x<=10; x++)**  **cout<<square(x)<<' ';**  **}**  **int square(int& y)**  **{ return y \* y; }** | | *Значение* фактического параметра **x** копируется в локальную переменную **y**, являющую формальным параметром. | *Адрес* фактического параметра **x** передается в указатель **y**. Для получения значения в функции пользователя требуется операция разыменования. | Параметр **y** выступает *псевдонимом* фактического параметра **x**, то есть передаётся сам объект. Нет необходимости в разыменовании указателя. |   Условие: Написать программу которая выводит квадраты чисел с 1по 10 |
| 2. Изучить использование *указателей* и *ссылок* как *формальных параметров* функции при работе с *одномерным* *массивом данных*, опробовав работу программ в правой части.  Написать условие задачи. | | #include <iostream>  using namespace std;  int sum(int x[]) // определение функции  {  int res = 0;  for (int i = 0; i < 5; i++)  res += x[i];  return res;  }  int main()  {  int mas[5], i;  for (i = 0; i < 5; i++)  {  mas[i] = i \* i;  cout << mas[i] << ' ';  }  cout << endl << "result=" << sum(mas) << endl;  }    **Условие: Создать массив из квадратов чисел с 0 по 4. И найти их сумму** |
| 3. В программе, записанной справа, используется меню, разработанное с помощью оператора **switch**, которое позволяет делать выбор между двумя функциями.  Написать условие задачи и комментарии к программе. | | #include <iostream>  using namespace std;  int massiv(int\* M, int); //прототип  void matrix(); //прототип  int main(void)  {  setlocale(LC\_CTYPE, "Russian");  int choice;  do  {  cout << "Выберите вариант работы" << endl;  cout << "1 - с одномерным массивом" << endl;  cout << "2 - с матрицей" << endl;  cout << "3 - выход" << endl;  cin >> choice;  switch (choice)  {  case 1: int\* M, size, i, av;  cout << "Введите размер массива ";  cin >> size;  M = new int[size];  for (i = 0; i < size; i++)  {  cout << "Введите " << i + 1 << " число ";  cin >> \*(M + i);  }//инициализирование и заполнение массива числами  av = massiv(M, size);//передача массива в функцию  cout << "Результат=" << av << endl;  delete[] M;  break;  case 2: matrix();//вызов функции для обработки матрицы  break;  case 3: break;  }  } while (choice != 3);  }  int massiv(int\* M, int size)//Функция которая считает сумму элементов массива  {  int sum = 0, avar;  for (int i = 0; i < size; i++)  sum += \*(M + i);  avar = sum / size;  return avar;  }  void matrix()//функция для работы с матрицей  {  int\*\* A, row, col, i, j, m;  cout << "Введите число строк матрицы ";  cin >> row;  cout << "Введите чиcло столбцов ";  cin >> col;  A = new int\* [row];  for (i = 0; i < row; i++)  A[i] = new int[col];  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  {  cout << "Введите A[" << i << "],[" << j << "]= ";  cin >> \*(\*(A + i) + j);  }//заполнение матрицы  m = A[0][0];//максимальный элемент  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  if (\*(\*(A + i) + j) > m)  m = \*(\*(A + i) + j);//поиск максимального элемента  cout << "Результат=" << m << endl;//вывод макс  for (i = 0; i < row; i++)  delete A[i];  delete[] A;//очистка динамической памяти  } |
| 4. В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов и ***функций* *пользователя*** для условий задач из таблицы.  Объединить написанные программы, разработав интерфейс с помощью оператора **switch**. Начальные размерности и значения элементов массивов ввести с клавиатуры в главной функции.  Массивы передать из главной функции в функции пользователя как ***параметры***. | | |
| №Варианта  10 | | |
| Задание | Код | |
| 1. Дан двумерный массив, состоящий из **N** строк и **М** столбцов, а также число **k**. Найти столбец, содержащий это число.  2. Дана строка слов, разделенных пробелами, в конце строки точка. Поменять местами два центральных слова, если их количество четно. | case 10: {  int part;  cout << "Введите номер задания:";  cin >> part;  switch (part) {  case 1: {  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];  }  }  int k;  cout << "Введите число которое необходимо найти:";  cin >> k;  int temp = Z10\_1(a, n, m, k);  if (temp == -1) {  cout << "Нет столбца с элементом k";  }  else {  cout << "Столбец номер:" << temp + 1;  }  for (int i = 0; i < n; i++) {  delete[] a[i];  }  delete[] a;  return 0;  }  case 2: {  int n;  cout << "Введите длину строки:";  cin >> n;  n++;  char\* s = new char[n];  cin.ignore(cin.rdbuf()->in\_avail());  cin.getline(s, n);  cout << '\n';  Z10\_2(s, n);  delete[] s;  return 0;  }  }  }  int Z10\_1(int\*\* a, int n, int m, int k) {  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (a[i][j] == k) {  return j;  }  }  }  return -1;  }  void Z10\_2(char\* s, int n) {  setlocale(LC\_ALL, "rus");  SetConsoleOutputCP(1251);  SetConsoleCP(1251);  int k = 0;  for (int i = 0; i < n; i++) {  if (s[i] == ' ') {  k++;  }  }  if (k % 2 == 0) {  cout << "Количество слов нечётно";  return;  }  k = (k - 1) / 2;  int start = -1,end=-1;  int m;  for (int i = 0; i < n; i++) {  if (s[i] == ' ') {  k--;  }  if (k > 0) {  continue;  }  if (start == -1)  start = i;  if (k == -1 && s[i] == ' ') {  m = i-start;  }  if ((k == -2 && s[i] == ' ') || s[i] == '.') {  end = i;  break;  }  }  char\* first = new char[m];  for (int i = start; i < start + m; i++) {  first[i - start] = s[i];  }  for (int i = start+1; i < start+(end-start-m); i++) {  s[i] = s[i + m];  }  for (int i = end - m; i < end; i++) {  s[i] = first[i - end + m];  }  cout << s;  return;  } | |

Допы

|  |  |
| --- | --- |
| №Варианта 3 | |
| Задание | Код/результат |
| 1. Найти сумму элементов, лежащих ниже главной диагонали в целочисленном массиве **A**[**n**, **n**].  2. Написать программу, которая вводит несколько строк текста и символ «**m**» и использует функцию, чтобы определить суммарное число вхождений символа в текст. | case 3: {  int part;  cout << "Введите номер задания:";  cin >> part;  switch (part){  case 1: {  int n;  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];  }  }  int sum = Z3\_1(a, n);  cout << "Сумма элементов ниже главной диагонали" << sum;  return 0;  }  case 2: {  int n;  cout << "Введите количество символов в тексте";  cin >> n;  char\* s = new char [n];  for (int i = 0; i < n;) {  char\* a = new char[n];  cin.ignore(cin.rdbuf()->in\_avail());  cin.getline(a, n);  int m = strnlen\_s(a, n);  for (int j = 0; j <= m; j++) {  s[i] = a[j];  i++;  }  }  char k;  cout << "Введите искомый символ:";  cin >> k;  int sum = Z3\_2(s, n, k);  cout << "Количество вхождений:" << sum;  return 0;  }  }  }  int Z3\_1(int\*\* a, int n) {  int sum = 0;  for (int i = 0; i < n; i++) {  for (int j = 0; j < i; j++) {  sum += a[i][j];  }  }  return sum;  }  int Z3\_2(char\* s, int n, char k) {  int sum = 0;  for (int i = 0; i < n; i++) {  if (s[i] == k)  sum++;  }  return sum;  } |

|  |  |
| --- | --- |
| №Варианта 8 | |
| Задание | Код/результат |
| 1. Дан двумерный массив, состоящий из **N** строк и **М** столбцов. Найти количество отрицательных, положительных и нулевых элементов массива.  2. Написать программу, которая вводит несколько слов текста и определяет общее количество слов. Слова разделены пробелами. | case 8: {  int part;  cout << "Введите номер задания:";  cin >> part;  switch (part){  case 1: {  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];  }  }  int\* ans = Z8\_1(a,n,m);  cout << "Количество положительных:" << ans[0] << '\n';;  cout << "Количество нулей:" << ans[1] << '\n';  cout << "Количество отрицательных:" << ans[2] << '\n';  return 0;  }  case 2: {  int n;  cout << "Введите количество символов:";  cin >> n;  char\* s = new char[n];  cin.ignore(cin.rdbuf()->in\_avail());  cin.getline(s, n);  int sum = Z8\_2(s, n);  cout << "Количество слов:" << sum + 1;  return 0;  }  }  }  void Z8\_1(int\*\* a, int n, int m, int ans[3]) {  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (a[i][j] > 0) {  ans[0]++;  }  if (a[i][j] == 0) {  ans[1]++;  }  if (a[i][j] < 0) {  ans[2]++;  }  }  }  }  int Z8\_2(char\* s, int n) {  int sum = 0;  for (int i = 0; i < n; i++) {  if (s[i] == ' ') {  sum++;  }  }  return sum;  } |
| №Варианта 12 | |
| Задание | Код/результат |
| 1. Ввести целые числа в массив **A**[**n**, **m**] (**n**, **m** > 0) и все отрицательные элементы в нечетных строках заменить на такие же положительные.  2. Написать программу, преобразующую строку, состоящую только из прописных букв, в строку, состоящую из прописных и строчных букв, при этом первая буква после точки и пробела − прописная, остальные − строчные. | case 1: {  int part;  cout << "Введите номер задания:";  cin >> part;  switch (part) {  case 1: {  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];  }  }  Z1\_1(a, n, m);  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cout << a[i][j] << ' ';  }  cout << '\n';  }  return 0;  }  case 2: {  int n;  cout << "Введите количество символов:";  cin >> n;  char\* s = new char[n];  cin.ignore(cin.rdbuf()->in\_avail());  cin.getline(s, n);  Z1\_2(s, n);  return 0;  }  }  }  void Z1\_1(int\*\* a, int n, int m) {  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (a[i][j] < 0 && i % 2 != 0) {  a[i][j] \*= -1;  }  }  }  }  void Z1\_2(char\* s, int n) {  for (int i = 1; i < n; i++) {  if (s[i - 1] == 0 || s[i-1]=='.') {  s[i] += 32;  }  }  } |