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***.

Варыянт 3

|  |
| --- |
| 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. |
| #include <iostream>  #include <Windows.h>  using namespace std;  //функцыя для знаходжання сумы элементаў пад галоўнай дыяганаллю  int sumBelowMainDiagonal(int\*\* ptr, int n) {  int result = 0;  //перабіраем элементы пад галоўнай дыяганаллю і знаходзім із суму  for (int i = 1; i < n; i++) {  for (int j = 0; j < i; j++) {  result += ptr[i][j];  }  }  //вяртаем суму  return result;  }  //функцыя для падліку колькасці сімвалаў symbol  int charsOccurrences(char\*\* ptr, int numLines, char symbol) {  int count = 0;  for (int i = 0; i < numLines; ++i) {  for (int j = 0; ptr[i][j] != '\0'; ++j) { //калі '\0', то радок скончыўся і далей не ідзем  if (ptr[i][j] == symbol) { //падлічваем  count++;  }  }  }  //вяртаем  return count;  }  int main()  {  SetConsoleOutputCP(1251);  //Выбар што рабіць  char choice;  //для 1 задання  int n, rmn = -10, rmx = 10;  int\*\* ptr;  //для 2 задання  char\*\* text;  int numLines;  char symbol;  cout << "Выберыце варыянт: \n";  cout << "1. Найти сумму элементов, лежащих ниже главной диагонали в целочисленном массиве A[n, n].\n";  cout << "2. Написать программу, которая вводит несколько строк текста и символ «m» и использует функцию, чтобы определить суммарное число вхождений символа в текст. \n";  cin >> choice;  switch (choice) {  case'1':  //Ствараем масіў n на n з дапамогай new  cout << "Увядзіце памер матрыцы: "; cin >> n;  ptr = new int\* [n];  for (int i = 0; i < n; ++i)  ptr[i] = new int[n];  srand((unsigned)time(NULL));  //Запаўняем масіў выпадковымі лікамі і адразу выводзім яго  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  ptr[i][j] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  cout << ptr[i][j] << " ";  }  cout << '\n';  }  //выклікаем функцыю  cout << sumBelowMainDiagonal(ptr, n);  break;  case'2':  cout << "Увядзіце колькасць радкоў: "; cin >> numLines;  cout << "Увядзіце сімвал(рэгістр улічваецца): "; cin >> symbol;  cin.ignore(); //каб пазбегнуць праблем з уводам калі можа застацца сімвал \n і будзе уведзена пусты радок  text = new char\* [numLines]; //ствараем масіў на numLines радкоў  for (int i = 0; i < numLines; ++i) {  text[i] = new char[256]; //256 сімвалаў павінна хапіць на сказ. карыстаюсь такім метадам, каб не карыстацца std::string, які мы не вывучалі😊  cout << "Увядзіце радок " << (i + 1) << ": ";  cin.getline(text[i], 256); //уводзім радок з дапамогай getline каб улічваць прабелы за сімвалы  }  cout << charsOccurrences(text, numLines, symbol); //выклікаем функцыю і выводзім вынік  break;  default:  cout << "няправільны варыянт";  return 0;  }  } |
|  |
|  |

Варыянт 2

|  |
| --- |
| 1. Ввести вещественные числа в массив **А**[**n**, **m**] (**n**, **m** > 0) и ко всем числам каждой строки прибавить единицу. Если первый элемент строки нулевой, ничего не делать.  **2.** Дана строка. Все русские буквы привести к верхнему регистру, латинские − заменить символом «**?**». Вывести результат на экран. |
| #include <iostream>  #include <Windows.h>  using namespace std;  //функцыя для павелічэння элементаў матрыцы на 1, адрозных ад 0  void decreaseMatrixValues(double\*\* ptr, int n, int m) {  int result = 0;  //перабіраем элементы  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (ptr[i][j] != 0) { //калі гэта не 0, то  ptr[i][j] = ptr[i][j] + 1; //павялічваем на 1  cout << ptr[i][j] << ' '; //выводзім  }  else {  cout << ptr[i][j] << ' '; //інакш толькі выводзім  }  }  cout << '\n';  }  }  //функцыя для змянення зыходнага сказа  void charsOccurrences(char\* ptr) {  int count = 0;  for (int i = 0; ptr[i] != '\0'; i++) {  if ('а' <= ptr[i] && ptr[i] <= 'я') { //калі гэта маленькая руская літара  ptr[i] = ptr[i] - 32; //робім яе вялікай шляхам адымання 32 ад яя(будзе аднята 32 ад кода, розніца ў якіх якраз у 32 паміж маленькай і вялікай)  cout << ptr[i];  }  else if ('A' <= ptr[i] && ptr[i] <= 'z') { //для ангельскіх літар  ptr[i] = '?'; //мяняем на знак пытання  cout << ptr[i];  }  else {  cout << ptr[i]; //інакш не мяняем  }  }  }  int main()  {  SetConsoleOutputCP(1251);  SetConsoleCP(1251);  //Выбар што рабіць  char choice;  //для 1 задання  int n, m, rmn = -10, rmx = 10;  double\*\* ptr;  //для 2 задання  char\* text;  cout << "Выберыце варыянт: \n";  cout << "1. Ввести вещественные числа в массив А[n, m] (n, m > 0) и ко всем числам каждой строки прибавить единицу. Если первый элемент строки нулевой, ничего не делать.\n";  cout << "2. Дана строка. Все русские буквы привести к верхнему регистру, латинские - заменить символом «?». Вывести результат на экран. \n";  cin >> choice;  switch (choice) {  case'1':  //Ствараем масіў n на m з дапамогай new  cout << "Увядзіце памер матрыцы n x m: "; cin >> n >> m;  ptr = new double\* [n];  for (int i = 0; i < n; i++)  ptr[i] = new double[m];  srand((unsigned)time(NULL));  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  ptr[i][j] = ((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn;  cout << ptr[i][j] << " ";  }  cout << '\n';  }  //выклікаем функцыю  cout << "Змененая матрыца: \n";  decreaseMatrixValues(ptr, n, m);  break;  case'2':  cin.ignore(); //каб пазбегнуць праблем з уводам калі можа застацца сімвал \n і будзе уведзена пусты радок  text = new char[256]; //ствараем масіў 256 сімвалаў  cin.getline(text, 256); //уводзім радок з дапамогай getline каб улічваць прабелы за  charsOccurrences(text); //выклікаем функцыю і выводзім вынік  break;  default:  cout << "няправільны варыянт";  return 0;  }  } |
|  |

Варыянт 5

|  |
| --- |
| 1. Дан двумерный массив.  В каждой строке выбрать минимальный элемент и среди них максимальный.  Напечатать номер строки, в которой расположен этот элемент.  2. Написать программу, которая вводит несколько строк текста и символ поиска и определяет суммарное число вхождений символа в текст |
| #include <iostream>  #include <Windows.h>  using namespace std;  //функцыя для знаходжання сумы элементаў пад галоўнай дыяганаллю  int MinMaxIndexFind(int\*\* ptr, int n, int m) {  int minV, maxV = ptr[0][0], minMaxIndex = 0;  //перабіраем элементы пад галоўнай дыяганаллю і знаходзім із суму  for (int i = 0; i < n; i++) {  minV = ptr[i][0];  for (int j = 0; j < m; j++) {  if (ptr[i][j] < minV) {  minV = ptr[i][j]; //калі элемент менш за мішімальны, то захоўваем яго як мінімальны  }  }  if (minV > maxV) { //калі мінімальны з гэтага радка больш за максімальны з ранейшых радкоў, то захоўваем яго як максімальны і захоўваем яго індэкс  minMaxIndex = i;  maxV = minV;  }  }  //вяртаем індэкс  return ++minMaxIndex;  }  //функцыя для падліку колькасці сімвалаў symbol  int charsOccurrences(char\*\* ptr, int numLines, char symbol) {  int count = 0;  for (int i = 0; i < numLines; ++i) {  for (int j = 0; ptr[i][j] != '\0'; ++j) {  if (ptr[i][j] == symbol) { //падлічваем  count++;  }  }  }  //вяртаем  return count;  }  int main()  {  SetConsoleOutputCP(1251);  //Выбар што рабіць  char choice;  //для 1 задання  int n,m, rmn = -10, rmx = 10;  int\*\* ptr;  //для 2 задання  char\*\* text;  int numLines;  char symbol;  cout << "Выберыце варыянт: \n";  cout << "1. Найти сумму элементов, лежащих ниже главной диагонали в целочисленном массиве A[n, n].\n";  cout << "2. Написать программу, которая вводит несколько строк текста и символ «m» и использует функцию, чтобы определить суммарное число вхождений символа в текст. \n";  cin >> choice;  switch (choice) {  case'1':  //Ствараем масіў n на n з дапамогай new  cout << "Увядзіце памер матрыцы n x m: "; cin >> n>>m;  ptr = new int\* [n];  for (int i = 0; i < n; ++i)  ptr[i] = new int[m];  srand((unsigned)time(NULL));  //Запаўняем масіў выпадковымі лікамі і адразу выводзім яго  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  ptr[i][j] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  cout << ptr[i][j] << " ";  }  cout << '\n';  }  //выклікаем функцыю  cout << MinMaxIndexFind(ptr, n, m);  break;  case'2':  cout << "Увядзіце колькасць радкоў: "; cin >> numLines;  cout << "Увядзіце сімвал(рэгістр улічваецца): "; cin >> symbol;  cin.ignore(); //каб пазбегнуць праблем з уводам калі можа застацца сімвал \n і будзе уведзена пусты радок  text = new char\* [numLines]; //ствараем масіў на numLines радкоў  for (int i = 0; i < numLines; ++i) {  text[i] = new char[256]; //256 сімвалаў павінна хапіць на сказ. карыстаюсь такім метадам, каб не карыстацца std::string, які мы не вывучалі😊  cout << "Увядзіце радок " << (i + 1) << ": ";  cin.getline(text[i], 256); //уводзім радок з дапамогай getline каб улічваць прабелы за сімвалы  }  cout << charsOccurrences(text, numLines, symbol); //выклікаем функцыю і выводзім вынік  break;  default:  cout << "няправільны варыянт";  return 0;  }  } |
|  |

Варыянт 7

|  |
| --- |
| 1. Дан двумерный массив, состоящий из **N** строк и **М** столбцов, а также число **d**. Найти строку, содержащую число **d**.  2. В строке подсчитать сумму кодов символов каждого слова. Слова в строке разделены пробелами. |
| #include <iostream>  #include <Windows.h>  using namespace std;  //функцыя для знаходжання сумы элементаў пад галоўнай дыяганаллю  void IndexOfTheRow(int\*\* ptr, int n, int m, int b) {  int minMaxIndex = 0;  //перабіраем элементы пад галоўнай дыяганаллю і знаходзім із суму  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (ptr[i][j] == b) {  cout << "Індэкс радка, дзе сустрэўся " << b << ": " << ++i;  return; //Выводзім інэдкс першага радка, дзе сустрэўся лік і выходзім з функцыі  }  }  }  //калі лік не быў знойдзены, то будзе выканан гэты код  cout << "Няма b у масіве";  }  //функцыя для падліку колькасці сімвалаў symbol  int countChars(char\* ptr) {  int count = 0;  for (int i = 0; ptr[i] != '\0'; i++) { //калі '\0', то радок скончыўся і далей не ідзем  if (ptr[i] != ' ') { //лічым кольксць усіх сімвалаў акрамя прабелаў  count++;  }  }  //вяртаем  return count;  }  int main()  {  SetConsoleOutputCP(1251);  //Выбар што рабіць  char choice;  //для 1 задання  int n, m, b, rmn = -10, rmx = 10;  int\*\* ptr;  //для 2 задання  char\* text;  cout << "Выберыце варыянт: \n";  cout << "1. Дан двумерный массив, состоящий из N строк и М столбцов, а также число d. Найти строку, содержащую число d. \n";  cout << "2. В строке подсчитать сумму кодов символов каждого слова. Слова в строке разделены пробелами. \n";  cin >> choice;  switch (choice) {  case'1':  //Ствараем масіў n на n з дапамогай new  cout << "Увядзіце памер матрыцы n x m: "; cin >> n >> m;  cout << "Увядзіце лік, які трэба знайсці: "; cin >> b;  ptr = new int\* [n];  for (int i = 0; i < n; ++i)  ptr[i] = new int[m];  srand((unsigned)time(NULL));  //Запаўняем масіў выпадковымі лікамі і адразу выводзім яго  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  ptr[i][j] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  cout << ptr[i][j] << " ";  }  cout << '\n';  }  //выклікаем функцыю  IndexOfTheRow(ptr, n, m, b);  break;  case'2':  cin.ignore(); //каб пазбегнуць праблем з уводам калі можа застацца сімвал \n і будзе уведзена пусты радок  text = new char[256]; //ствараем масіў 256 сімвалаў  cin.getline(text, 256); //уводзім радок з дапамогай getline каб улічваць прабелы за  countChars(text);  //выклікаем функцыю і выводзім вынік  cout << countChars(text);  break;  default:  cout << "няправільны варыянт";  return 0;  }  } |
|  |