|  |  |
| --- | --- |
| #include <iostream>  **26 катя**  using namespace std;  int main() {  const int maxInputSize = 1000;  // Ввод строки с произвольным текстом  cout << "Enter string: ";  char input[maxInputSize];  cin.getline(input, maxInputSize);  // Вывод слов, содержащих сим'  cout << "Words with 'a' 'A': ";  bool insideWord = false;  char currentWord[maxInputSize];  int wordIndex = 0;  int countWords = 0;  int wordLongIndex = -1;  int wordLength = 0;  for (int i = 0; i < maxInputSize && input[i] != '\0'; ++i) {  if (input[i] == ' ' || input[i] == '\t') {  // Завершаем текущее слово  if (insideWord) {  insideWord = false;  currentWord[wordIndex] = '\0';  countWords++;  if ((currentWord[0] == 'a') || (currentWord[0] == 'A')) {  cout << currentWord << " ";  }  if (wordIndex > wordLength) {  wordLongIndex++;  wordLength = wordIndex;  }  //Очищаю массив с уже существующим словом  for (int j = 0; j < wordIndex; j++) {  currentWord[j] = ' ';  }  wordIndex = 0;  }  }  else {  // Добавляем символ к текущему слову  insideWord = true;  currentWord[wordIndex++] = input[i];  }  }  // Проверяем последнее слово в строке  if (insideWord && ((currentWord[0] == 'a') || (currentWord[0] == 'A'))) {  cout << currentWord;  }  //Проверка последнего слова на длину  if (wordIndex > wordLength) {  wordLongIndex++;  }  cout << endl << "Position of the longest word is " << wordLongIndex+1 << endl;  return 0;  } | #include <iostream>  **29 яна**  #include <cstring>  using namespace std;  bool isLatinLetter(char c)  {  return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z');  }  void matching\_letters(const char\* inputString) {  int length = strlen(inputString);  int wordCount = 0;  int matchingFirstAndLastCharCount = 0;  int fiveLetterWordCount = 0;  const char\* wordStart = nullptr;  for (int i = 0; i < length; i++) {  if (isLatinLetter(inputString[i])) {  // Начало слова  if (wordStart == nullptr) {  wordStart = &inputString[i];  /\* wordCount++;\*/  }  }  else if (wordStart != nullptr) {  // Конец слова  const char\* wordEnd = &inputString[i - 1];  if (\*wordStart == \*wordEnd) {  matchingFirstAndLastCharCount++;  }  int wordLength = wordEnd - wordStart + 1;  if (wordLength == 5) {  fiveLetterWordCount++;  }  wordStart = nullptr;  }  }  // Обработка последнего слова, если оно есть  if (wordStart != nullptr) {  const char\* wordEnd = &inputString[length - 1];  if (\*wordStart == \*wordEnd) {  matchingFirstAndLastCharCount++;  }  int wordLength = wordEnd - wordStart + 1;  if (wordLength == 5) {  fiveLetterWordCount++;  }  }  cout << "Количество слов с совпадающей первой и последней буквой: " << matchingFirstAndLastCharCount << endl;  cout << "Количество слов из 5 букв: " << fiveLetterWordCount << endl;  /\*cout << "Общее количество слов: " << wordCount << endl;\*/  }  int main() {  setlocale(LC\_ALL, "russian");  const int MAX\_SIZE = 100; // Максимальная длина строки  char inputString[MAX\_SIZE];  cout << "Введите строку: ";  cin.getline(inputString, MAX\_SIZE);  matching\_letters(inputString);  return 0;  } |
|  | int main() {  **22никита**  setlocale(LC\_ALL, "Russian");  int N;  cout << "Введите размерность матрицы: ";  cin >> N;  int\*\* A = new int\* [N];  for (int i = 0; i < N; i++) {  A[i] = new int[N]; }  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << "Введите элемент A[" << i << "][" << j << "]: ";  cin >> \*(A[i] + j); } }  int product = 1;  int nonzeroCount = 0;  for (int i = 0; i < N; i++) {  bool allPositive = true;  for (int j = 0; j < N; j++) {  if (\*(A[i] + j) <= 0) {  allPositive = false;  break;  } }  if (allPositive) {  product = 0;  }  else {  product = 1; } }  cout << "Исходная матрица A:" << endl;  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << \*(A[i] + j) << " "; }  cout << endl; }  cout << "Произведение положительных элементов: " << product << endl;  cout << "Количество ненулевых элементов: " << nonzeroCount << endl;  for (int i = 0; i < N; i++) {  delete[] A[i]; } delete[] A; return 0; } |
| #include <iostream> **2никита**  int main() {  setlocale(LC\_ALL, "Russian");  // Получение размера массива от пользователя  int N;  cout << "Введите размер массива: ";  cin >> N;  // Динамическое выделение памяти для массива A  int\* A = new int[N];  // Инициализация массива A  for (int i = 0; i < N; i++) {  cout << "Введите элемент A[" << i << "]: ";  cin >> A[i];  }  // Вывод исходного массива A  cout << "Исходный массив A:" << endl;  for (int i = 0; i < N; i++) {  cout << A[i] << " ";  }  cout << endl;  // Вычисление суммы положительных элементов и увеличение отрицательных элементов на 4  int sumPositive = 0;  for (int i = 0; i < N; i++) {  if (A[i] > 0) {  sumPositive += A[i];  }  else if (A[i] < 0) {  A[i] += 4;  }  }  cout << "Сумма положительных элементов: " << sumPositive << endl;  // Удаление элементов, у которых сумма соседних элементов больше 12  for (int i = 1; i < N - 1; i++) {  if (A[i - 1] + A[i + 1] > 12) {  for (int j = i; j < N - 1; j++) {  A[j] = A[j + 1];  }  N--;  i--;  }  }  cout << "Измененный массив A:" << endl;  for (int i = 0; i < N; i++) {  cout << A[i] << " ";  }  cout << endl;  delete[] A;  return 0;} | int main() { **3никита**  int N;  cout << "Введите размер матрицы: ";  cin >> N;  int\*\* A = new int\* [N];  for (int i = 0; i < N; i++) {  A[i] = new int[N];  }  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << "Введите элемент A[" << i << "][" << j << "]: ";  cin >> A[i][j];  }  }  cout << "Исходная матрица A:" << endl;  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << A[i][j] << " ";  }  cout << endl;  }  int sumNegative = 0;  int productEven = 1;  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  if (A[i][j] < 0) {  sumNegative += A[i][j];  }  if (A[i][j] % 2 == 0) {  productEven \*= A[i][j]; } } }  int\* B = new int[N];  for (int k = 0; k < N; k++) {  B[k] = 1;  for (int i = 0; i < N; i++) {  if (A[i][k] % 2 != 0) {  B[k] = 1;  break;  }  }  }  cout << "Массив B:" << endl;  for (int k = 0; k < N; k++) {  cout << B[k] << " ";  }  cout << endl;  for (int i = 0; i < N; i++) {  delete[] A[i]; } delete[] A; delete[] B; return 0;} |
| #include <iostream**> 25 или 27??? яхз**  #include <string.h>  using namespace std;  void printWordsWithLength4AndSNotF(const char\* str, const int max\_size) {  int i = 0;  char\* word\_with\_s\_no\_f = new char[max\_size];  while (str[i] != '\0') {  // Пропускаем пробелы  while (str[i] == ' ') {  i++;  }  // Считываем слово  char \*word=new char[max\_size];  char\* word\_with\_four= new char[max\_size];  char\* word\_with\_s\_no\_f = new char[max\_size];  int j = 0;  while (str[i] != ' ' && str[i] != '\0') {  word[j] = str[i];  i++;  j++;  }  word[j] = '\0';  // Проверяем условия для вывода слова  if (j == 4 ) {  strcat(word\_with\_four, word);  }  if (strchr(word, 's') != nullptr && strchr(word, 'f') == nullptr) {  strcat(word\_with\_s\_no\_f, word);  }  cout << "Words with 4 letters: " << word\_with\_four << endl;  cout << "Words with 's' and no 'f': " << word\_with\_s\_no\_f << endl;  delete word\_with\_s\_no\_f;  delete word;  }  }  int main() {  setlocale(LC\_ALL, "ru");  const int max\_size = 100;  cout << "Enter the string^ ";  char str[max\_size];  cin.getline(str, max\_size);  printWordsWithLength4AndSNotF(str, max\_size);  return 0;  }  #include <iostream**> 7никита**  int main() {  int N;  cout << "Введите размер массива: ";  cin >> N;  int\* A = new int[N];  for (int i = 0; i < N; i++) {  cout << "Введите элемент A[" << i << "]: ";  cin >> A[i];  }  cout << "Исходный массив A:" << endl;  for (int i = 0; i < N; i++) {  cout << A[i] << " ";  }  cout << endl;  int sumEven = 0;  for (int i = 0; i < N; i++) {  if (A[i] % 2 == 0) {  sumEven += A[i];  }  if (A[i] < 0) {  A[i] = 0;  }  }  for (int i = 1; i < N - 1; i++) {  if (A[i] == 0) {  A[i] = (A[i - 1] + A[i + 1]) / 2;  }  }  cout << "Сумма четных элементов: " << sumEven << endl;  cout << "Измененный массив A:" << endl;  for (int i = 0; i < N; i++) {  cout << A[i] << " ";  }  cout << endl;  delete[] A;  return 0;  } | #include <iostream**> 19 даша**  using namespace std;  void func(int\* a, int n) {  int sum = 0;  int below = 0;  for (int i = 0; i < n; i++) {  if (a[i] < 0) sum += a[i];  if (a[i] % 2 == 0) a[i] += 4;  cout << a[i] << "\t";  if (a[i] < 0) below++;  }  cout << endl;  cout << "sum = " << sum << endl;  int\* b = new int[n];  for (int i = 0, k = 0; i < n; i++) {  if (a[i] < 0) {  b[k] = a[i];  k++;  }  else {  b[below] = a[i];  below++;  }  }  for(int i = 0; i < n; i++)  cout << b[i] << "\t";  cout << endl;  }  int main() {  int n;  cin >> n;  int\* a = new int[n];  for (int i = 0; i < n; i++) {  cin >> a[i];  }  for (int i = 0; i < n; i++) {  cout << a[i] << "\t";  }  cout << endl;  func(a, n);  }  #include <iostream>  **10никита**  int main() {  int N;  cout << "Введите размер матрицы: ";  cin >> N;  int\*\* A = new int\*[N];  for (int i = 0; i < N; i++) {  A[i] = new int[N];  }  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << "Введите элемент A[" << i << "][" << j << "]: ";  cin >> A[i][j];  }  }  cout << "Исходная матрица A:" << endl;  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << A[i][j] << " ";  }  cout << endl;  }  int negativeCount = 0;  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  if (A[i][j] < 0) {  negativeCount++;  }  }  }  int row = 1; // Индекс второй строки (индексы начинаются с 0)  int sumRow = 0;  for (int j = 0; j < N; j++) {  sumRow += A[row][j];  }  double averageRow = static\_cast<double>(sumRow) / N;  int minColumn = 0;  int maxColumn = 0;  int minValue = A[0][0];  int maxValue = A[0][0];  for (int j = 1; j < N; j++) {  for (int i = 0; i < N; i++) {  if (A[i][j] < minValue) {  minValue = A[i][j];  minColumn = j;  }  if (A[i][j] > maxValue) {  maxValue = A[i][j];  maxColumn = j;  }  }  }  // Обмен значениями между столбцами  for (int i = 0; i < N; i++) {  int temp = A[i][minColumn];  A[i][minColumn] = A[i][maxColumn];  A[i][maxColumn] = temp;  }  cout << "Количество отрицательных элементов: " << negativeCount << endl;  cout << "Среднее арифметическое элементов второй строки: " << averageRow << endl;  // Вывод измененной матрицы A  cout << "Измененная матрица A:" << endl;  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << A[i][j] << " ";  }  cout << endl;  }  for (int i = 0; i < N; i++) {  delete[] A[i];  }  delete[] A;  return 0;  } |
| **билет 19 катя хз**  #include <iomanip>  #include <iostream>  #include <cmath>  using namespace std;  double y(double x) {  return (2 \* pow(x, 2) + 1) \* exp(pow(x, 2));  }  long double fac(int n) {  if(n == 0) return 1;  return n \* fac(n - 1);  }  long double s(double x, int N) {  long double result = 0;  for(int m = 0;m <= N;m++) {  result += ((2\*m + 1)\*pow(x, 2\*m)/fac(m));  }  return result;  }  int main() {  double x = 0.2;  double step = 0.135;  double end = 1.2;  int N = 1000;  cout << "x\t\t\t\t\t\ty(x)\t\ts(x,N)" << endl;  cout << "-----------------------------------------------" << endl;  while (x <= end) {  cout << fixed << setprecision(10);  cout << x << "\t\t" << y(x) << "\t" << s(x, N) << endl;  x += step;  }  return 0;  } |
| int main() { **15никита**  int N;  cout << "Введите размер матрицы: ";  cin >> N;  int\*\* A = new int\* [N];  for (int i = 0; i < N; i++) {  A[i] = new int[N];  }  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << "Введите элемент A[" << i << "][" << j << "]: ";  cin >> A[i][j];  }  }  cout << "Исходная матрица A:" << endl;  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << A[i][j] << " ";  }  cout << endl;  }  int\* B = new int[N];  for (int j = 0; j < N; j++) {  int sum = 0;  for (int i = 0; i < N; i++) {  sum += A[i][j];  }  int average = sum / N;  if (average > 7) {  B[j] = 0;  }  else {  B[j] = 1;  }  }  cout << "Массив B:" << endl;  for (int i = 0; i < N; i++) {  cout << B[i] << " ";  }  cout << endl;  // Замена элементов четных строк на 0  for (int i = 0; i < N; i += 2) {  for (int j = 0; j < N; j++) {  A[i][j] = 0;  }  }  cout << "Матрица A с замененными четными строками:" << endl;  for (int i = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  cout << A[i][j] << " ";  }  cout << endl;  }  for (int i = 0; i < N; i++) {  delete[] A[i];  }  delete[] A;  delete[] B;  return 0;  } | #include <iostream**> 28 я хз мб вообще неандестендабле 1 часть и то не працуе**  #include <string.h>  using namespace std;  //bool checkWord(const char\* word) {  // int length = strlen(word);  // if (length < 4) {  // return false;  // }  // if (word[0] == word[length - 1]) {  // return false;  // }  // if (word[1] == word[2]) {  // return true;  // }  // return false;  //}  void findAndPrintWords(const char\* input\_str) {  const char\* delimiter = " ";  const char\* wordStart = input\_str;  const char\* wordEnd = nullptr;  while (\*input\_str) {  if (\*input\_str == ' ') {  if (wordStart != nullptr && wordEnd != nullptr) {  int wordLength = wordEnd - wordStart;  char\* word = new char[wordLength + 1];  strncpy(word, wordStart, wordLength);  word[wordLength] = '\0';  if (strstr(word, "www") != nullptr /\*&& checkWord(word)\*/) {  cout << word << endl;  }  delete[] word;  wordStart = nullptr;  wordEnd = nullptr;  }  }  else {  if (wordStart == nullptr) {  wordStart = input\_str;  }  wordEnd = input\_str + 1;  }  ++input\_str;  }  if (wordStart != nullptr && wordEnd != nullptr) {  int wordLength = wordEnd - wordStart;  char\* word = new char[wordLength + 1];  strncpy(word, wordStart, wordLength);  word[wordLength] = '\0';  if (strstr(word, "www") != nullptr /\*&& checkWord(word)\*/) {  cout << word << endl;  }  delete[] word;  }  }  int main() {  const int max\_size = 1000;  char str[max\_size];  cout << "Введите строку: ";  cin.getline(str, max\_size);  findAndPrintWords(str);  return 0;  } |
| #include <iostream> **БИЛЕТ 22 даша**  using namespace std;  void func(int\* a, int n) {  int count\_1 = 0;  int big\_num = a[0];  for (int i = 0; i < n; i++) {  int count = 0;  int num = a[i];  for (int j = 0; j < n; j++) {  if (a[j] == num && i == 0) count\_1++;  if (a[j] == num && i > 0) count++;  }  if (i > 0 && count > count\_1) {  count\_1 = count;  big\_num = num;  }  }  cout << "Самый частый эл-т массива: " << big\_num << endl;  int min = a[0];  int max = a[0];  for (int i = 0; i < n; i++) {  if (a[i] < min) min = a[i];  if (a[i] > max) max = a[i];  if (n - i <= 4) a[i] = 7;  cout << a[i] << " ";  }  cout << endl;  cout << "min = " << min << endl;  cout << "max = " << max << endl;  }  int main() {  setlocale(LC\_ALL, "RU");  int n;  cout << "Ведите кол-во эл-ов массива: ";  cin >> n;  int\* a = new int[n];  cout << "Введите эл-ты массива: ";  for (int i = 0; i < n; i++) {  cin >> a[i];  }  cout << endl;  cout << "Массив: ";  for (int i = 0; i < n; i++) {  cout << a[i] << " ";  }  cout << endl;  func(a, n);  delete[] a;  return 0;  } | #include <iostream> **10 даша**  using namespace std;  void func(int\*\* a, int n) {  int count = 0;  double arifm = 0;  int min\_i = 0, max\_i = 0;  int min = a[0][0];  int max = a[0][0];  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if (a[i][j] < 0) count++;  if (i == 1) arifm += a[i][j];  if (a[i][j] > max) {  max = a[i][j];  max\_i = j;  }  if (a[i][j] < min) {  min = a[i][j];  min\_i = j;  }  }  }  arifm /= n;  cout << "count = " << count << endl;  cout << "arifm = " << arifm << endl;  int\* buffer = new int[n];  for (int i = 0; i < n; i++) {  buffer[i] = a[i][min\_i];  a[i][min\_i] = a[i][max\_i];  a[i][max\_i] = buffer[i];  }  delete[] buffer;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << a[i][j] << "\t";  }  cout << endl;  }  }  int main() {  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];  }  }  cout << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << a[i][j] << "\t";  }  cout << endl;  }  func(a, n);  for (int i = 0; i < n; i++)  delete[] a[i];  delete[] a;  return 0;  } |
| #include <iostream> **6 билет даша**  #include <math.h>  #include <iomanip>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  double x, s;  int i;  for (x = 0.3; x < 1.7 + 0.093 / 2; x += 0.093)  {  double r = 1; s = 0;  for (int i = 1; i < 1100; i++) {  r \*= (-1) \* (2 \* x) \* (2 \* x) / ((2 \* i) \* (2 \* i - 1));  s += r;  }  cout << setw(15) << x << setw(15) << 2 \* (cos(x) \* cos(x) - 1) << setw(15) << s << endl;  }  return 0;  }  **Билет 5** #include <iostream>  #include <iomanip>  using namespace std;  int main() {  setlocale(LC\_ALL, "russian");  int n;  cout << "Введите количество столбцов(строк) матрицы: ";  cin >> n;  int\*\* mat = new int\* [n];  cout << "Введите элементы матрицы: " << endl;  for (int i = 0; i < n; i++) {  mat[i] = new int [n];  for (int j = 0; j < n; j++) {  cin >> mat[i][j];  }  }  int pro, count, nepol,otr\_count;  pro = 1;  count =nepol= 0;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if (mat[i][j] > 0) {  pro \*= mat[i][j];  }  else {  nepol++;  }  if(mat[i][j]>=0) count++;  }  }  cout << "Your matrix: " << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << mat[i][j]<<" ";  }  cout << endl;  }  if (nepol < n \* n) {  cout << endl << "Произведенеие положительных элементов: " << pro << endl;  }  else {  cout << "В матрице отсутствуют положительные элементы! "<<endl;  };  if (count == 0) {  cout << "Все элементы матрицы отрицательные!" << " Количество ненулевых элементов матрицы: " << count << endl;;  }  else {  cout << "Количество ненулевых элементов матрицы: " << count << endl;  };  int\* B = new int[n]();  cout << "Массив В: " << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n ; j++) {  if (mat[i][j] <= 0) {  B[i] = 1;  break;  }  else {  B[i]=0;  }  }  cout << setw(2) << B[i] << setw(2);  }  cout << endl;  delete[] B;  for (int i = 0; i < n; i++) {  delete mat[i];  }  delete mat;  return 0;  } | #include <iostream> **билет31**  using namespace std;  int main() {  const int maxInputSize = 1000; // Максимальный размер ввода  // Ввод строки с произвольным текстом  cout << "Enter string: ";  char input[maxInputSize];  cin.getline(input, maxInputSize);  // Вывод слов, содержащих символ 'w'  cout << "Words with 'w': ";  bool insideWord = false;  char currentWord[maxInputSize];  char addWord[maxInputSize];  char lastWord[maxInputSize];  int lastnum = 0;  int numLastWord = 0;  int wordIndex = 0;  int countWords = 0;  //Определяем позицию последнего символа  for (int i = 0; i < maxInputSize; i++) {  if (input[i] == '\0') break;  lastnum++; }  for (int i = lastnum-1; i >= 0; i--) {  if (input[i] == ' ' || input[i] == '\t' || (i == -1)) {  if (insideWord) {  insideWord = false;  lastWord[wordIndex] = '\0';  countWords++;  if (countWords == 2) {  //cout << wordIndex << endl;  break;  }  //Удаляю ненужное слово  for (int j = 0; j <= wordIndex; j++) {  lastWord[j] = ' ';  }  wordIndex = 0;  numLastWord = 0;  }  }  else {  insideWord = true;  lastWord[wordIndex++] = input[i];  numLastWord++; }  }  wordIndex = 0;  countWords = 0;  for (int i = 0; i < maxInputSize && input[i] != '\0'; ++i) {  if (input[i] == ' ' || input[i] == '\t') {  // Завершаем текущее слово  if (insideWord) {  insideWord = false;  currentWord[wordIndex] = '\0';  countWords++;  if (strchr(currentWord, 'w') != nullptr) {  cout << currentWord << " ";  }  //Заношу второе слово в другой массив  if (countWords == 2) {  for (int j = 0; j < wordIndex; j++) {  addWord[j] = currentWord[j];  }  }  //Очищаю массив с уже существующим словом  for (int j = 0; j < wordIndex; j++) {  currentWord[j] = ' ';  }  wordIndex = 0;  }  }  else {  // Добавляем символ к текущему слову  insideWord = true;  currentWord[wordIndex++] = input[i];  }  }  // Проверяем последнее слово в строке  if (insideWord && strchr(currentWord, 'w') != nullptr) {  cout << currentWord; }  cout << endl;  if (numLastWord == 3) {  cout << "The penultimate word: ";  for (int i = numLastWord - 1; i >= 0; i--) {  cout << lastWord[i];  }  cout << endl;  }  else {  cout << addWord; }  return 0;} |
| #include <iostream> ,**bktn 30**  using namespace std;  int main() {  const int maxInputSize = 1000; // Максимальный размер ввода  // Ввод строки с произвольным текстом  cout << "Enter string: ";  char input[maxInputSize];  cin.getline(input, maxInputSize);  // Вывод слов, содержащих символ 'w'  cout << "Words with 's': ";  bool insideWord = false;  char currentWord[maxInputSize];  int EndedBY[maxInputSize / 2];  int EndedBYIndex = 0;  int wordIndex = 0;  int countWords = 0;  for (int i = 0; i < maxInputSize; ++i) {  if (input[i] == ' ' || input[i] == '\0') {  // Завершаем текущее слово  if (insideWord) {  insideWord = false;  currentWord[wordIndex] = '\0';  countWords++;  if (currentWord[0] == 's') {  cout << currentWord << " ";  }  if (currentWord[wordIndex - 2] == 'b' && currentWord[wordIndex - 1] == 'y') {  EndedBY[EndedBYIndex++] = countWords;  }  //Очищаю массив с уже существующим словом  for (int j = 0; j < wordIndex; j++) {  currentWord[j] = ' ';  }  wordIndex = 0;  if (input[i] == '\0') break;  }  }  else {  // Добавляем символ к текущему слову  insideWord = true;  currentWord[wordIndex++] = input[i];  }  }  cout << "\nIndexes of word, which ended at 'by' :";  for (int i = 0; i < EndedBYIndex; i++) {  cout << EndedBY[i] << ' ';  }  return 0;  } | #include <iostream>  **11**  #include <cmath>  #include <iomanip>  using namespace std;  double y(double x) {  return (1 - (x\*x / 2)) \* cos(x) - (x / 2) \* sin(x);  }  double s(double &x, int N) {  double result = 0;  for (int m = 0; m <= N; m++) {  double sum = pow(-1, m) \* (2 \* m\*m + 1) \* pow(x, 2 \* m);  double fact = 1;  for (int i = 1; i <= 2 \* m; i++) {  fact \*= i;  }  result += sum / fact;  }  return result;  }  int main() {  setlocale(LC\_ALL, "ru");  cout << "Таблица значений ряда у(х):" << endl;  cout << "x"<<setw(20) <<"y(x)" << setw(20)<<"s(x)" << endl;  for (double x = 0.2; x <= 1.4; x += 0.043) {  cout << x << setw(20) << y(x)<<setw(20)<<s(x,820) << endl;  }  return 0;  } |