Вариант 3

Задание 4

Написать и выполнить программу по первой блок-схеме лабораторной работы № 2.

|  |
| --- |
| Программный код |
| #include <iostream> // Для использования cout, cin  using namespace std; // Чтобы не писать каждый раз std  int main()  {  setlocale(LC\_CTYPE, "Russian"); // Для корректного отображения символов русской раскладки  double x, y, z; // Объявление переменных  // Запрос на ввод чисел для инициализации переменных пользователем  cout << "Введите число x: "; cin >> x;  cout << "Введите число y: "; cin >> y;  cout << "Введите число z: "; cin >> z;  // Вычисление суммы и произведения чисел и запись результата в соответствующие переменные  double sum = x + y + z, product = x \* z \* y;  // Проверяем больше ли сумма произведения, если да, то выводим ее утроенное значение, иначе утроенное произведение  if (sum > product) {  cout << sum \* 3;  }  else {  cout << product \* 3;  }  } |
| Вывод программы |
|  |

Задание 5

Даны три числа **a**, **b** и **c**. Найти среднее геометрическое этих чисел, если все они отличны от нуля, и среднее арифметическое в противном случае.

|  |
| --- |
| Программный код |
| #include <iostream> // Для использования cout, cin  using namespace std; // Чтобы не писать каждый раз std  int main()  {  setlocale(LC\_CTYPE, "Russian"); // Для коректного отображения символов русской раскладки  double a, b, c; // Объявление переменных  // Запрос на ввод чисел для инициализации переменных пользователем  cout << "Введите число x: "; cin >> a;  cout << "Введите число y: "; cin >> b;  cout << "Введите число z: "; cin >> c;  // Проверяем отличны ли все числа от 0, если да, то вычисляем и выводим среднее геометрическое этих чисел, иначе среднее арифметическое  if (a != 0 && b != 0 && c != 0) {  double result = sqrt(a \* b \* c); // Вычисляем среднее геометрическое  cout <<result ;  }  else {  double result = (a + b + c) / 3; // Вычисляем среднее арифметическое  cout <<result;  }  } |
| Вывод программы |
|  |

Задание 6

|  |
| --- |
| Программный код |
| #include <iostream>  using namespace std; // Чтобы не писать каждый раз std  #include <Windows.h> // Для использования SetConsoleOutputCP(1251) и SetConsoleCP(1251)  int main()  {  // Для корректного отображения русских символов в консоли  SetConsoleOutputCP(1251);  SetConsoleCP(1251);  int answer; // Объявление переменной для ответа пользователя  // Вывод сообщения и чтение ответа пользователя  puts("Добро пожаловать в \"Мы открылись\", что будете брать? (1 - кофе, 2 - бургер, 3 - пака)");  cin >> answer;  // Инструкции для выбора продуктов  switch (answer) {  case 1: // Если пользователь выбрал кофе  puts("Хотите взять к кофе булочку с повидлом? У нас акция, всего за 3.99 кофе + булочка (1 - да, 2 - нет)");  cin >> answer;  switch (answer) {  case 1: puts("Отлично! Ожидайте"); break;  case 2: puts("Хорошо, вот ваш кофе"); break;  }  break;  case 2: // Если пользователь выбрал бургер  puts("Хотите взять напиток? (1 - да, 2 - нет)");  cin >> answer;  switch (answer) {  case 1: // Если пользователь хочет напиток  puts("Какой будете брать? (1 - спрайт, 2 - квас, 3 - кола)");  cin >> answer;  switch (answer) {  case 1: puts("Ожидайте, ваш спрайт и бургер скоро будут готовы"); break;  case 2: puts("Ожидайте, ваш квас и бургер скоро будут готовы"); break;  case 3: puts("Ожидайте, ваша кола и бургер скоро будут готовы"); break;  }  break;  case 2: puts("Ожидайте!"); break;  }  break;  case 3: puts("Ну и пока"); break;  // При некоректном вводе  default: puts("Не понял вас"); break;  }  } |
| Вывод программы |
|  |

Доп. Задания:

Задание 6. На шахматной доске стоят черный король и белые ладья и слон. Проверить, есть ли угроза королю и если есть, то от кого именно

|  |
| --- |
| Программный код |
| #include <iostream>  using namespace std;  // Функция преобразования символа в целое число (от 0 до 7).  // Например, 'a' будет преобразовано в 0, 'b' в 1, и так далее.  int convertCharToInt(char columnChar) {  int columnInt = abs(columnChar - 'a' - 7);  return columnInt;  }  // Функция проверки видит ли слон короля.  bool bishopIsBlocked(int kingX, int kingY, int bishopX, int bishopY, int rookX, int rookY, int A[8][8]) {  //Тут мы определяем, находится ли король выше или ниже слона, и исходя из этого будет выбран шаг для X и Y(отрицательный или положительный) с которым мы будем перебирать все клеточки между королем и слоном и смотреть, есть ли между ними ладья  int deltaX = bishopX - kingX;  int deltaY = bishopY - kingY;  int stepX = (deltaX > 0) ? 1 : -1;  int stepY = (deltaY > 0) ? 1 : -1;  if (abs(deltaX) == abs(deltaY)) {  int x = kingX + stepX;  int y = kingY + stepY;  // Проверяем, есть ли между королем и слоном ладья.  while (x != bishopX && y != bishopY) {  if (A[y][x] == 1) {  return true; // Слон заблокирован  }  x += stepX;  y += stepY;  }  }  return false; // Слон не заблокирован  }  // Функция проверки, может ли ладья блокировать короля.  bool rookIsBlocked(int kingX, int kingY, int bishopX, int bishopY, int rookX, int rookY, int A[8][8]) {  if (kingX == rookX) {  // Определяем направление движения ладьи по вертикали.  int stepY = (kingY < rookY) ? 1 : -1;  int y = kingY + stepY;  // Проверяем, есть ли между королем и ладьей другие фигуры.  while (y != rookY) {  if (A[y][kingX] == 2) {  return true; // Ладья блокирует короля.  }  y += stepY;  }  }  else if (kingY == rookY) {  // Определяем направление движения ладьи по горизонтали.  int stepX = (kingX < rookX) ? 1 : -1;  int x = kingX + stepX;  // Проверяем, есть ли между королем и ладьей другие фигуры.  while (x != rookX) {  if (A[kingY][x] == 2) {  return true; // Ладья блокирует короля.  }  x += stepX;  }  }  return false; // Ладья не блокирует короля.  }  // Функция проверки, находится ли король под угрозой шаха от слона.  bool isKingAttackedByBishop(int kingY, int kingX, int bishopY, int bishopX, int rookX, int rookY, int A[8][8]) {  if (abs(kingX - bishopX) == abs(kingY - bishopY)) { // Если слон на одной диагонали с королем  if (!bishopIsBlocked(kingX, kingY, bishopX, bishopY, rookX, rookY, A)) { // Если  return true; // Король под атакой от слона.  }  }  return false; // Король не под атакой от слона.  }  // Функция проверки, находится ли король под угрозой шаха от ладьи.  bool isKingAttackedByRook(int kingY, int kingX, int rookY, int rookX, int bishopX, int bishopY, int A[8][8]) {  if (kingX == rookX || kingY == rookY) { // Если ладья на одной вертикали или горизонтали с королем  if (!rookIsBlocked(kingX, kingY, bishopX, bishopY, rookX, rookY, A)) { //Если слон не блокирует ладью  return true; // Король под атакой от ладьи.  }  }  return false; // Король не под атакой от ладьи.  }  int main() {  setlocale(LC\_ALL, "RU");  char kingXInCharType, rookXInCharType, bishopXInCharType;  int kingY, rookY, bishopY;  cout << "Введите позицию короля в формате a2, f5, h8 и так далее: ";  cin >> kingXInCharType >> kingY;  cout << "Введите позицию ладьи: ";  cin >> rookXInCharType >> rookY;  cout << "Введите позицию слона: ";  cin >> bishopXInCharType >> bishopY;  // Преобразование внутренних индексов доски в индексы массива.  kingY -= 1;  bishopY -= 1;  rookY -= 1;  int kingX = convertCharToInt(kingXInCharType),  rookX = convertCharToInt(rookXInCharType),  bishopX = convertCharToInt(bishopXInCharType);  int A[8][8] = { 0 }; // Создание матрицы 8 на 8 для представления шахматной доски, где 0 - пустая клеточка, 1 - ладья, 2 - слон  A[rookY][rookX] = 1; // Маркировка ладьи как 1.  A[bishopY][bishopX] = 2; // Маркировка слона как 2.  // Проверка, находится ли король под угрозой от ладьи и/или слона, результат которой присваевается переменной  bool checkByRook = isKingAttackedByRook(kingY, kingX, rookY, rookX, bishopX, bishopY, A);  bool checkByBishop = isKingAttackedByBishop(kingX, kingY, bishopX, bishopY, rookX, rookY, A);  if (checkByBishop) {  cout << "Шах от слона!" << endl;  }  if (checkByRook) {  cout << "Шах от ладьи!" << endl;  }  if (!checkByBishop && !checkByRook) {  cout << "Король в безопасности." << endl;  }  return 0;  } |
| Тест 1 |
|  |
| Тест 2 |
|  |
| Тест 3 |
|  |
| Тест 4 |
|  |
| Тест 5 |
|  |

Задание 1. Введены с клавиатуры три числа. Найти сумму тех чисел, которые делятся на 5. Если таких чисел нет, то выве-сти текст «Error». Использовать только тернарные операторы.

|  |
| --- |
| Программный код |
| #include <iostream>  using namespace std; // Чтобы не писать каждый раз std  #include <Windows.h> // Для использования SetConsoleOutputCP(1251)  int main()  {  SetConsoleOutputCP(1251); // Устанавливаем кодировку консоли для корректного отображения кириллицы  int a, b, c, result = 0; // Объявление переменных  // Вводим три числа с клавиатуры  cout << "Введите первое число: "; cin >> a;  cout << "Введите второе число: "; cin >> b;  cout << "Введите третье число: "; cin >> c;  // Используем тернарные операторы для проверки и суммирования чисел, делящихся на 5  result += (a % 5 == 0) ? a : 0;  result += (b % 5 == 0) ? b : 0;  result += (c % 5 == 0) ? c : 0;  // Проверяем, были ли числа, делящиеся на 5  (result > 0) ? cout << result : cout << "Error";  return 0;  } |
| Вывод программы |
|  |

Задание 4. Пройдет ли шар радиуса r через ромбообразное отверстие с диагоналями p и q?

|  |
| --- |
| Программный код |
| #include <iostream>  using namespace std; // Чтобы не писать каждый раз std  #include <Windows.h> // Для использования SetConsoleOutputCP(1251)  int main()  {  SetConsoleOutputCP(1251); // Устанавливаем кодировку консоли для корректного отображения кириллицы  double r, p, q; // Объявление переменных  // Ввод переменных через пробел  cout << "Введите радиус шара и значения двух диагоналей ромба через пробел: "; cin >> r >> p >> q;  //Если радиус вписанной в ромб окружности больше или равен радиусу шара, то шар пройдет  if ((0.5 \* sqrt(pow(p / 2, 2)) + pow(q / 2, 2)) >= r) {  cout << "Да"; // Выводим да, если шар проходит  }  else {  cout << "Нет"; //Выводим нет, если шар не проходит  }  return 0;  } |
| Вывод программы |
|  |