Лабораторная №7

Задание 1

Код моей программы из лаб.6(задание 5(2))

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_CTYPE, "Russian");  float b = 0.5, m = 8, a = 20, y, z, j = 2;  while (j<=3)  {  y = (m - b) / (sin(a) - exp(a));  z = 3 \* y + sqrt(a - 4 \* j \* b);  cout << "y = " << y << "\t" << "z = " << z << endl;  j += 0.5;  }  return 0;  }  Прокрутка:  b = 0.5, m = 8, a = 20, y, z, j = 2;  2<=3; y = (m - b) / (sin(a) - exp(a));z = 3 \* y + sqrt(a - 4 \* j \* b); вывод y=-1.54587e-08 и z=4  j =j+0.5=2+0.5=2.5  2.5<=3; y = (m - b) / (sin(a) - exp(a));z = 3 \* y + sqrt(a - 4 \* j \* b); вывод y=-1.54587e-08 и z=3.87298  j =j+0.5=2.5+0.5=3  3<=3; y = (m - b) / (sin(a) - exp(a));z = 3 \* y + sqrt(a - 4 \* j \* b); вывод y=-1.54587e-08 и z=3.74166  j =j+0.5=3+0.5=3.5  3.5 > 3. Выполнение программы закончено |

Задание 2

Я убрал строчку x = x + 2 и тогда цикл выполняется бесконечно

|  |
| --- |
| #include <stdio.h>  void main()  {  float y, x = 0;  while (x < 7)  {  if (x < 4)  y =x;  else  if (x > 4)  y = x \* x;  else  y = 1;  printf("x=%f\t", x);  printf("y=%5.2f\n", y);  }  } |

Задание 3

В нашем примере шаг с обходом (F10) и шаг с заходом (F11) не отличаются, так как не используются ни библиотечные функции, ни пользовательские. В случае, если бы такие функции присутствовали, шаг с обходом (F10) не заходил бы внутрь функции, а просто выполнял её и принимал возвращаемое значение. Шаг с заходом (F11) позволил бы войти в функцию и просмотреть её внутреннюю реализацию.

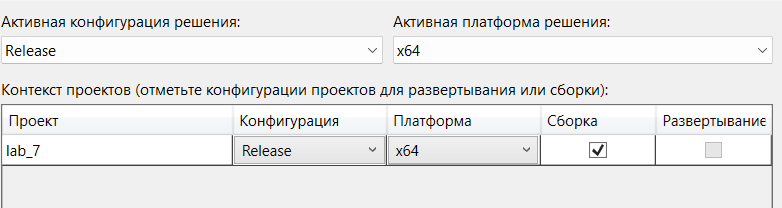
|  |
| --- |
| #include <stdio.h>  void main()  {  float y, x = 0;  while (x < 7)  {  if (x < 4)  y = x;  else  if (x > 4)  y = x \* x;  else  y = 1;  printf("x=%f\t", x);  printf("y=%5.2f\n", y);  x += 2;  }  } |

Задание 4

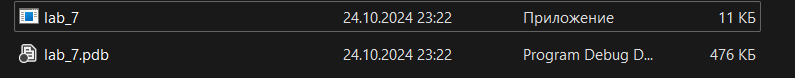
|  |
| --- |
|  |

Задание 5

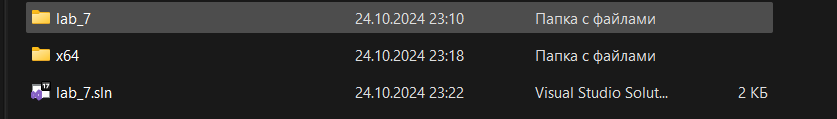




Release:



Объяснить назначение папок и файлов решения проекта.

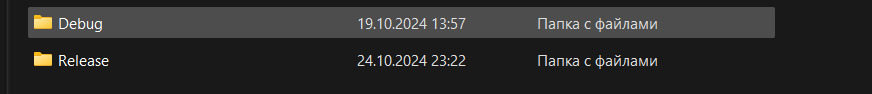


* В папке lab\_7 содержатся исходные файлы, заголовочные файлы и другие ресурсы для проекта



1. **x64 (папка)** — это папка, которая содержит результаты сборки для архитектуры x64.
2. **lab\_7.cpp** — это файл с исходным кодом программы на языке C++.
3. **lab\_7.vcxproj** — это файл проекта Visual Studio, который содержит настройки сборки, пути к исходным и заголовочным файлам, а также другие конфигурационные параметры для этого конкретного проекта.
4. **lab\_7.vcxproj.filters** — этот файл определяет фильтры которые используются для организации файлов проекта внутри Visual Studio. Это может быть, например, распределение по категориям, таким как "Файлы исходного кода" или "Заголовочные файлы".
5. **lab\_7.vcxproj.user** — это файл настроек пользователя, который хранит информацию о пользовательских настройках проекта, таких как настройки для отладки или компиляции.

* **В папке x6**4 содержит результаты сборки для архитектуры x64. Здесь могут находиться папки Debug и Release, в которых хранятся скомпилированные файлы, такие как .exe или .dll.



* **lab\_7.sln (файл)** — это файл решения Visual Studio с расширением .sln, который объединяет в себе настройки для всех проектов, входящих в это решение. Этот файл управляет тем, как проекты внутри решения взаимодействуют друг с другом, а также хранит ссылки на проекты и их настройки.

Задание 6 ( вариант 12)

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  int main() {  double x, y;  double c = 3.7;  double d = 51.9e-2;  double a = 4;  double k = 1;  while (c < 5)  {  x = tan(pow(a, 2) - 1) / (d + 1);  if (3 \* x < a \* c) {  y = a \* k + d;  }  else {  y = cos(a \* k) \* exp(a + 1);  }  cout << "x = " << x << endl;  cout << "y = " << y << endl;  c += 0.1;  }  return 0;  } |

Задание 7

|  |
| --- |
| #include <iostream>  using namespace std;  // используем алгоритм Евклида  int gcd(int a, int b) {  while (a != b) {  if (a > b) {  a = a - b;  }  else {  b = b - a;  }  }  return a;  }  int gcdOfThree(int x, int y, int z) {  return gcd(gcd(x, y), z);  }  int main() {  setlocale(LC\_ALL, "ru");  int x, y, z;  cout << "Введите три натуральных числа: \n";  cin >> x >> y >> z;  cout << "НОД чисел " << x << ", " << y << " и " << z << " = " << gcdOfThree(x, y, z) << endl;  return 0;  } |

Доп.задания:

Задание 3

|  |
| --- |
| Найти натуральное число, состоящее из трёх цифр, с возрастающими слева направо цифрами, являющееся полным квадратом. Число является полным квадратом, если квадратный корень из него – простое число (число 121 – полный квадрат, т. к. 121=11 ∙ 11, а 11 – простое число). |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  // проверка простого числа  bool prost(int n) {  if (n < 2)  return false;  for (int i = 2; i <= sqrt(n); i++) {  if (n % i == 0)  return false;  }  return true;  }  bool voz(int num) {  int lastDigit = num % 10; // последняя цифра  int midDigit = (num /= 10) % 10; // средняя цифра  int firstDigit = num / 10; // первая цифра  return (firstDigit < midDigit && midDigit < lastDigit);  }  int main() {  setlocale(LC\_ALL, "ru");  for (int num = 100; num <= 999; num++) {  int sqrtNum = sqrt(num);  // тут проверка, является ли число полным квадратом и простым  if (sqrtNum \* sqrtNum == num && prost(sqrtNum)) {  // тут проверка, возрастают ли цифры числа  if (voz(num)) {  cout << "Число: " << num << " является полным квадратом простого числа." << endl;  }  }  }  return 0;  } |

Задание 2

|  |
| --- |
| Составить алгоритм, определяющий, сколько существует способов набора одного рубля при помощи монет достоинством 50 коп., 20 коп., 5 коп. и 2 коп. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "ru");  int count = 0;  // цикл по монетам 50 копеек  for (int a = 0; a <= 1; a++) { // максимум 1 монета по 50 коп.  // цикл по монетам 20 копеек  for (int b = 0; b <= 5; b++) { // максимум 5 монет по 20 коп.  // цикл по монетам 5 копеек  for (int c = 0; c <= 20; c++) { // максимум 20 монет по 5 коп.  // цикл по монетам 2 копейки  for (int d = 0; d <= 50; d++) { // максимум 50 монет по 2 коп.  // проверка, равна ли сумма 100 копейкам  if (a \* 50 + b \* 20 + c \* 5 + d \* 2 == 100) {  count++;  }  }  }  }  }  cout << "Количество способов набора одного рубля: " << count << endl;  return 0;  } |

Задание 4

|  |
| --- |
| Три приятеля были свидетелями нарушения правил дорожного движения. Номер автомобиля – четырехзначное число – никто полностью не запомнил. Из показаний следует, что номер делится на 2, на 7 и на 11, в записи номера участвуют только две цифры, сумма цифр номера равна 30. Составить алгоритм и программу для определения номера автомашины. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "ru");  for (int num = 1000; num <= 9999; num++) {  if (num % 2 == 0 && num % 7 == 0 && num % 11 == 0) {  int digits[10] = { 0 }; // массив для подсчета встречаемости каждой цифры  int sum = 0; // переменная для хранения суммы цифр  int tempNum = num; // Временная переменная для работы с числом  // разбиваем число на цифры и считаем сумму  while (tempNum > 0) {  int digit = tempNum % 10; // получаем последнюю цифру  digits[digit]++; // увеличиваем счетчик для этой цифры  sum += digit; // добавляем цифру к сумме  tempNum /= 10; // убираем последнюю цифру  }  // сколько различных цифр встречается  int uniqueDigits = 0;  for (int i = 0; i < 10; i++) {  if (digits[i] > 0) {  uniqueDigits++; // Увеличиваем счетчик уникальных цифр  }  }  if (uniqueDigits == 2 && sum == 30) {  cout << "Найденный номер автомобиля: " << num << endl;  }  }  }  return 0;  } |