**Лабораторная работа № 7. Отладка программ**

|  |  |  |  |
| --- | --- | --- | --- |
| Задание | | Решение | |
| 1. Записать условие задачи для программы, представленной в правой части.  Выполнить прокрутку для одной из программ лабораторной работы № 6. | | **#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 = x + 2;**  **}**  **}**  Вывести значение функции y для x=0(2)6  #include <iostream>  #include <cmath>  using namespace std;  int main() {  double s = 7.4, m = 10, f = 3.2e4;  double k[3] = { 4,0.5,8 };  for (int i = 0; i < 3; i++) {  double y = s / log(5.2 \* f) / (exp(-s) + 2);  double v = (1 + m \* y - m \* k[i]) / log(y);  cout << "y=" << y << endl;  cout << "v=" << v << endl;  }  return 0;  }  s = 7.4; m = 10; f = 3.2e4; k = {4, 0.5, 8};  i = 0; k[i] = 4;  y = s / log(5.2 \* f) / (exp(-s) + 2) = 1.419; v = (1 + m \* y - m \* k[i]) / log(y) = -3.660; вывод y = 1.419; v = -3.660;  i = 1; k[i] = 0.5; y = s / log(5.2 \* f) / (exp(-s) + 2) = 1.419; v = (1 + m \* y - m \* k[i]) / log(y) = 27.032; вывод y = 1.419; v = 27.032;  i = 2; k[i] = 8; y = s / log(5.2 \* f) / (exp(-s) + 2) = 1.419; v = (1 + m \* y - m \* k[i]) / log(y) = -37.544; вывод y = 1.419; v = -37.544;  i = 3; i >= 3. Выполнение программы закончено. | |
| 2. Выполнить программу из п. 1 с использованием средства *интерактивной* отладки **Шаг с обходом**.  Внести изменения в программу с тем, чтобы появились ошибки (например, заменить == на = или убрать строчку **x = x + 2;** и т. п.). Проверить, как будет выполняться отладка по шагам с обходом. | | #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);  }  }  Приводит к бесконечному циклу(Если убрать x=x+2) | |
| 3. Выполнить программу из п. 1 с использованием средства *интерактивной* отладки **Шаг с заходом**.  Проанализировать изменения, происходящие в процессе отладки. | | Изменений нет, т.к. в коде нет функций. | |
| 4. Выполнить программу п. 1 с использованием средств *планируемой* отладки, установив точки останова на операторах вывода и пошагово проследив вывод значений на консоль.  Удалить точки останова. | |  | |
| 5. Создать исполняемый файл без отладочной информации **Release**. Объяснить назначение папок и файлов решения проекта.  Выполнить файл с расширением **\*.exe** в пап-ке **Debug**. | |  | |
| 6. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства ***интерактивной*** отладки **Шаг с обходом**. Вызвать окно **Контрольное значение** **1**, поместить туда имена переменных и проконтролировать их значения в процессе выполнения программы.  В отчете представить результаты в окне **Отладчика** (окно **Видимые** иокно **Контрольное значение** **1**).  Для той же программы представить результаты в окнах **Отладчика** для интерактивной отладки **Шаг с заходом** | | | |
| № Варианта | Задание | Код | Результат |
| 10 |  | #include <iostream>  using namespace std;  int main() {  double a = 1.055, x = 0.6;  for (int n = 6; n <= 14; n += 2) {  double y = pow(cos(x \* x / abs(x)), 2);  double z = y < (a \* x \* n) ? abs(y) : sqrt(1 + exp(-y));  cout << "y=" << y << ' ';  cout << "z=" << z << endl;  }  return 0;  } |  |

7. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства ***планируемой*** отладки.

В отчете представить результаты в окне **Отладчика** (окно **Локальные** и окно **Контрольное значение 1**).

Создать ***исполняемый файл*** и ознакомиться с файлами в папках проекта

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| № Варианта | Задание | Код | Результат |
| 10 | Используя перебор значений, найти сумму целых положительных чисел, кратных 5 и меньших 150 | #include <iostream>  using namespace std;  int main() {  int sum = 0;  for (int i = 0; i < 150; i++) {  if (i % 5 == 0) {  sum += i;  }  }  cout << sum;  return 0;  } |  |

|  |  |
| --- | --- |
| Задание | Код/результат |
| 1. 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). | for (int i = 1; i \* i < 1000; i++) {  if (i \* i < 99) continue;  if (!isprime(i)) continue;  int a, b, c;  a = i \* i;  c = a % 10;  a /= 10;  b = a % 10;  a /= 10;  if (a < b && b < c) cout << i \* i << endl;  }  return 0;  bool isprime(int a) {  for (int i = 2; i < a / 2; i++) {  if (a % i == 0) return false;  }  return true;  } |
| 2. Составить алгоритм, определяющий, сколько существует способов набора одного рубля при помощи монет достоинством 50 коп., 20 коп., 5 коп. и 2 коп. | int a = 100;  cout << ans(a, 0);  return 0;  int ans(int a, int k) {  int sum = 0;  if (a == 0) return 1;  for (int i = k; i < 4; i++) {  if (a < coin[i]) continue;  sum += ans(a - coin[i], i);  }  return sum;  } |
| 3. Имеются два сосуда. В первом сосуде находится C1 литров воды, во втором – C2 литров воды. Из первого сосуда переливают половину воды во второй сосуд, затем из второго переливают половину в первый сосуд, и т. д. Сколько воды окажется в обоих сосудах после 12 переливаний? | double C1, C2;  cin >> C1 >> C2;  for (int i = 0; i < 12; i++) {  if (i % 2 == 0) {  C2 += C1 / 2;  C1 /= 2;  }  else {  C1 += C2 / 2;  C2 /= 2;  }  }  cout << "C1=" << C1 << "л" << endl;  cout << "C2=" << C2 << "л" << endl;  return 0; |