**Лабораторная работа № 7. Отладка программ**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 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;**  **}**  **}**  **Условие**: Написать алгоритм выводящий на экран значения x и y пока x<7, при условии если x<4, тогда x=y, если x>4 тогда y=x^2, иначе y=1. x=0(2)7  **#include <iostream>**  **void main()**  **{**  **setlocale(LC\_CTYPE, "Russian");**  **float z, y, x, sd;**  **sd = 1.0 / 3.0;**  **for (int n = 0; n < 3; n++)**  **{**  **printf("Введите x ");**  **scanf\_s("%f", &x);**  **z = 2 \* pow(x, 2);**  **y = z + pow(x, sd);**  **printf("x = %5.2f\t", x);**  **printf("y = %5.2f\n", y);**  **}**  **}**  n=0  0<4; ввод x; z=2\*x^2; y=z+x^1/3; Вывод x и y; n+1;  1<4; ввод x; z=2\*x^2; y=z+x^1/3; Вывод x и y; n+1;  2<4; ввод x; z=2\*x^2; y=z+x^1/3; Вывод x и y; n+1;  3=3. Выполнение программы закончилось. |
| 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);**  **}**  **}**    Цикл выполняется бесконечно |
| 3. Выполнить программу из п. 1 с использованием средства *интерактивной* отладки **Шаг с заходом**.  Проанализировать изменения, происходящие в процессе отладки. | В данном случае процесс отладки выполняется точно так же, т.к. для шага с заходом нет вызываемых библиотек, пользовательских функцций и т.д. |
| 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.  6. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства ***интерактивной*** отладки **Шаг с обходом**  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.**  7. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже.  В отчете представить результаты в окне **Отладчика** (окно **Локальные** и окно **Контрольное значение 1**). | Внешние зависимости – содержит ссылки на все модули, которые  использует программа.  Файлы заголовков – содержит файлы кода С++ с расширением .h    Исходные файлы – содержит файлы кода С++ с расширением .срр    Файлы ресурсов – содержит файлы, непосредственно не относящиеся к  языку С++, но необходимые для работы приложения. Например,  мультимедийные файлы.  **#include <iostream>**  **using namespace std;**  **int main() {**  **setlocale(LC\_ALL, "RU");**  **double a = 1.33e-3, k = 6, x, d, b;**  **for (int i = 0; i < 4; ++i) {**  **cout << "Введите x: ";**  **cin >> x;**  **d = (exp(-x) + (x - a)) / log(x);**  **if (d >= k \* sqrt(a)) {**  **b = 6 \* d \* x;**  **}**  **else {**  **b = sin(a / x);**  **}**  **cout << "d= " << d << endl;**  **cout << "b= " << b << endl;**  **}**  **}**        **#include <iostream>**  **using namespace std;**  **int main() {**  **setlocale(LC\_ALL, "ru");**  **int n;**  **cout << "Введите натуральное число n: ";**  **cin >> n;**  **if (n <= 0) {**  **cout << "Число должно быть натуральным!" << endl;**  **return 1;**  **}**  **cout << "Натуральные делители числа " << n << ": ";**  **for (int i = 1; i <= n; i++) {**  **if (n % i == 0) {**  **cout << i << " ";**  **}**  **}**  **cout << endl;**  **return 0;**  **}** |
| **Доп. задания**  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 ways = 0;  for (int a = 0;a <= 2;a++) {  for (int b = 0;b <= 4;b++) {  for (int c = 0;c <= 20;c++) {  for (int d = 0;d <= 50;d++) {  if ((a \* 50 + b \* 20 + c \* 5 + d \* 2) == 100)  {  ways++;  }  }  }  }  }  cout << "Способов:" << ways << endl;  cout << clock() << "ms";  } |
| 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? | #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  double s1, s2;  cout << "Введите s1 и s2:"<<endl;  cin >> s1 >> s2;  for (int i = 0;i != 6;i++) {  s2 += s1 / 2;  s1 /= 2;  s1 += s2 / 2;  s2 /= 2;  }  cout << "s1=" << s1 << " L s2=" << s2<<" L";  } |