Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа №7

**Отладка программ**

Выполнил: Мышковец Артём Витальевич 10ПИ

Минск 2024

|  |  |
| --- | --- |
| **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);**  **}**  **}**  Программа из лаб. Работы №6  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. | 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);  }  } |
| 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 не изменился т.к. шагу с заходом некуда заходить(нет вызываемых библиотечных или пользовательских функций)  #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;  }  } |
| 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**. | 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. |
| 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**.  **Вариант 13** | #include <iostream>  using namespace std;  int main() {  double t, x, b = 3, i = 8, c = 6e-4, a = 6;  while (i < 24) {  t = (a \* i) / ((a \* a - b) \* exp(-a));  if (t > 5 \* c) {  x = 4.8e-3 + i \* a;  }  else {  x = a + i \* i \* t;  }  i += 4;  }  } |
| 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.  **Вариант 13**  Вывести первые 6 натуральных чисел, делителями которых являются числа 3 и 5. | #include <iostream>  using namespace std;  int main() {  int counter = 0, i = 1;  while (counter != 6) {  if (i % 3 == 0 && i % 5 == 0) {  counter++;  cout << i << endl;  }  i++;  }  } |

**Дополнительные задания**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 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). | #include <iostream>  using namespace std;  int main() {  int num = 100; //Начальное число равно 100 т.к. искомое число трехзначное  bool isSimple = true; //Переменная отвечает за состояние простоты числа  while (num<1000) {  int digits = num % 10; //Извлекаем Сотни, десятки и единицы из числа num  int tens = (num % 100 - digits)/10;  int hunderts = (num - tens - digits)/100;  if (hunderts < tens && tens < digits && int(sqrt(num))==sqrt(num)) { //Если соблюдается условие увеличение цифры в разряде слева-направо и корень является целочисленным тогда запускаем цикл проверки на простоту  for (int i = 2; i < sqrt(num);i++) { //Цикл для проверки делится ли число на какое-либо число вплоть до своего корня(i=2 т.к. любое число делится на 1)  if (int(sqrt(num)) % i == 0) { //Если нашлось число, на которое делится num без остатка, тогда оно не простое  isSimple = false; // меняем состояние  }  }  if (isSimple) { // Если после выполнения цикла проверки состояние не изменилось, тогда мы нашли искомое число и прерываем цикл  break;  }  }  num++; //Увеличиваем число на 1  }  cout << num << endl;  } |
| 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";  } |