**Лабораторная работа №1**

УСЛОВНЫЕ ОПЕРАТОРЫ И ЦИКЛИЧЕСКИЕ СТРУКТУРЫ В ЯЗЫКЕ СИ. ОПЕРАТОРЫ IF, IF-ELSE, SWITCH, FOR, WHILE, DO-WHILE

1. Цель работы

Изучить процесс прохождения программы на языке СИ в среде программирования. Ознакомиться с технологией отладки программ. Изучить правила оформления и использования условных операторов и операторов цикла в языке СИ

2. Краткие теоретические сведения

Microsoft Visual Studio – интегрированная среда разработки, которая позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms. Она также подходит для создания веб-сайтов, веб-приложений и веб-служб для всех поддерживаемых платформ: Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone, .NET Compact Framework и Silverlight. Visual Studio включает в себя редактор проектов, текстовый редактор, компиляторы с языков С, С++, компоновщик, средства отладки программ и библиотеки функций, программных и графических компонентов.

Вход в среду выполняется через меню «ПУСК»-«Microsoft Visual Studio 2017(22)», после чего для выполнения лабораторных работ данного курса создается новый консольный проект командами «Файл»-«Создать»-«Проект». В открывшемся окне «Visual C++» выбирается приложение «Консольное приложение Windows», при необходимости, указывается имя проекта путь к нему. После нажатия кнопки «OK» в текстовом редакторе открывается файл c именем «Имя Проекта».cpp, содержащий пустую функцию main, внутрь которой вводится текст программы.

Запуск программы: CTRL+F5 или меню «Отладка» – «Запуск без отладки». Отладка программы: F5 или меню «Отладка» – «Запустить отладку». В окне обозревателя решений можно добавлять файлы и управлять ими. В окне "Выходные данные" можно просматривать выходные данные сборки и другие сообщения. Справочная информация о среде программирования, библиотечных функциях и операторах языка может быть получена через команду меню «Справка» или нажатием клавиши клавиатуры CTRL+F1.

В процессе работы для ввода-вывода следует использовать функции printf и scanf. Основные форматы, используемые в этих функциях:

%d - ввод/вывод чисел типа int;

%o - то же, но в восьмеричном виде;

%f - ввод/вывод чисел типа float и double;

%с - ввод/вывод одного символа (char);

%s - ввод/вывод строки (массив char);

Дополнительно при выводе данных после знака % можно указывать длину поля вывода и количество знаков после запятой (только для %f). Например:

%5d - вывод целого числа в поле длиной 5 позиции;

%7.3f - вывод числа float в поле 7 позиций, 3 знака после запятой;

%20s - вывод строки в поле 20 позиций.

Данные при выводе выравниваются по правой границе. Для выравнивания по левой границе после «%» дополнительно указы¬вается знак "-", например «%-7.3f».

В форматной строке можно использовать управляющие константы:

\n - новая строка;

\t - табуляция;

\а - звуковой сигнал;

\b - возврат на шаг.

В общем случае условный оператор ***if*** в языке СИ записывается в виде

***if (выражение) оператор1;***

***else оператор2;***

или

***if (выражение) оператор1;***

Проверяется значение ***выражения***. Если оно не равно 0, выполняется ***оператор1***, в противном случае ***оператор2***. Далее в любом случае выполняется следующий оператор.

Оператор ***switch*** имеет вид:

***switch (выражение)***

***{***

***case конст1: оператор1;***

***case конст2: оператор2;***

***case констЗ: оператор3;***

***…***

***default: оператор4;***

***}***

Значение ***выражения*** в ***switch*** поочередно сравнивается с константами ***конст1***, ***конст2***, ***конст3***. В случае равенства выполняется соответствующий ***оператор*** (группа операторов) до конца ***switch*** или до оператора ***break***. Если ни одна из констант не равна значению выражения, выполняются операторы, начиная с метки ***default***. Если ***default*** отсутствует, не выполняется никаких действий.

Тернарная условная операция совмещает в себе операций проверки условия и присваивания. Она записывается в виде

***(выражение)? выражение1 : выражение2;***

Если ***выражение*** не равно 0, операция возвращает значение ***выражения1***, иначе ***выражения2***.

В языке СИ возможны три основных конструкции операторов цикла.

Оператор ***for*** имеет вид:

***for (оператор1; выражение; оператор2) оператор;***

например:

***for (i=0; i<=100; i++) a[i]=0;***

***Оператор1*** выполняется перед началом цикла (инициализация). ***Выражение*** является условием продолжения цикла. Если оно не равно 0, выполняется ***оператор*** (тело цикла). Затем выполняется ***оператор2*** (модификация). Процесс повторяется до тех пор, пока ***выражение*** не станет равным 0.

Оператор ***whilе*** имеет вид:

***while (выражение) оператор;***

например:

***while (x > 0) s += a[x];***

В этом случае ***оператор*** повторяется до тех пор, пока ***выражение*** не равно 0 (условие истинно). Проверка условия производится до выполнения оператора.

Цикл ***do-while*** имеет вид:

***do оператор;***

***while (выражение);***

например:

***do***

***{***

***s[i]=0; i++;***

***}***

***while (i < 5);***

Такая конструкция отличается от вышеописанной только тем, что значение ***выражения*** проверяется после выполнения ***оператора*** (тела цикла).

Во всех трех конструкциях циклов ***операторы*** могут быть составными.

Для выхода из цикла до его нормального завершения используется оператор ***break***. Он передает управление оператору, следующему за циклом. Оператор ***continue*** вызывает переход к началу цикла, т.е. к следующей итерации.

3. Методические указания

В лабораторной работе используются все вышеописанные конструкции циклов. В табл. 1.1. указано, какие из них вы должны использовать при выполнении тех или иных пунктов задания.

При написании программы с циклическими алгоритмами рекомендуется использовать фигурные скобки для явного указания тела цикла. Это уменьшает вероятность ошибок и делает понятной структуру программы.

При вводе строк используйте функцию ***gets()/gets\_s()***, так как функция ***scanf()*** воспринимает пробелы в строке как разделители.

Вывод строки на экран удобно выполнять функцией ***puts()***.

4. Порядок выполнения работы

Написать программу, выполняющую следующие действия:

4.1. Ввести две строки текста с подсказками при вводе;

4.2. Определить и вывести на экран длины строк (без конечного нулевого байта);

4.3. Объединить две строки в одну, удалив все пробелы. Для модификации использовать одну из условных операций;

4.4. Выполнить обработку строки (см. табл. 1.1). Обработку прекратить, если количество изменений будет больше четырех. Использовать для реализации задания конструкцию if-else.

4.5. Вывести результирующую строку на экран.

4.6. Повторить программу, применив для выполнения задания оператор switch в п.2. задания 4.4. Сравнить результаты.

Таблица 1.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Варианты  циклов | | | | Задание п. 4.4 |
| вар | п.  4.2 | п.  4.3 | | п. 4.4 |  |
| 1 | 1 | 2 | | 3 | 1. Во всех парах одинаковых символов второй заменить на пробел.  2. Заменить буквы A, M, P, R на цифры 0, 1, 2, 3 соответственно. |
| 2 | 1 | 3 | | 2 | 1. Во всех парах одинаковых символов первый заменить на «+».  2. Заменить буквы a, b, x, z на буквы e, k, m, p соответственно. |
| 3 | 2 | 1 | | 3 | 1. Заменить все пары букв «XY» на цифры «12».  2. Заменить все круглые и фигурные скобки на квадратные. |
| 4 | 2 | 3 | | 1 | 1. Заменить в строке все цифры пробелами.  2. Заменить знаки арифметических операций (+, -, \*, /) на буквы р, m, u, r. |
| 5 | 3 | 1 | | 2 | 1. Заменить на пробел все символы, совпадающие с последним символом в строке.  2. Заменить цифры 2, 4, 6, 3 на 9, 7, 5, 8 соответственно. |
| 6 | 3 | 2 | | 1 | 1. Заменить все символы с кодами меньше 48 на пробелы.  2. Заменить все квадратные и фигурные скобки на круглые. |
| 7 | 1 | 2 | | 3 | 1. Заменить все символы с кодами больше 48 на пробелы.  2. Заменить знаки $, %, &, # на U, R, L, X соответственно. |
| 8 | 1 | 3 | | 2 | 1. Заменить каждый третий символ на пробел.  2. Заменить буквы a, r, h, w на соответствующие верхнего регистра A, R, H, W. |
| 9 | 2 | 1 | | 3 | 1. Заменить все пары цифр «89» на буквы «AB».  2. Заменить цифры 1, 3, 6, 7 на буквы A, R, N, E соответственно. |
| 10 | 2 | 3 | | 1 | 1. Заменить все пары одинаковых символов на пробелы.  2. Заменить буквы а, б, л, н на f, g, u, k соответственно. |
| 11 | 3 | 1 | | 2 | 1. Заменить все четные символы в строке на пробелы.  2. Заменить цифры 3, 6, 9, 7 на знаки @, #, &, + соответственно. |
| 12 | 3 | 2 | 1 | | 1. Заменить все нечетные символы в строке на пробелы.  2. Заменить знаки <, >, !, ? на буквы f, t, x, z соответственно. |
| 13 | 1 | 2 | 3 | | 1. Заменить на пробелы все символы, совпадающие с первым символом в строке.  2. Заменить символы \, \_, ^, ~ на буквы q, w, e, r. |
| 14 | 1 | 3 | 2 | | 1. Заменить каждый третий символ в строке первым символом.  2. Заменить буквы q, s, x, v на буквы A, F, K, N соответственно. |
| 15 | 2 | 1 | 3 | | 1. Удалить из текстовой строки все четные символы.  2. Заменить буквы D, L, G, S на буквы k, f, t, r соответственно. |
| 16 | 2 | 3 | 1 | | 1. Удалить из текстовой строки все нечетные символы.  2. Заменить буквы s, y, j, k на цифры 4, 8, 1, 6 соответственно. |
| 17 | 3 | 1 | 2 | | 1. Удалить из текстовой строки все символы «,».  2. Заменить символы :, ;, -, + на буквы S, F, V, N соответственно. |
| 18 | 3 | 2 | 1 | | 1. Удалить из текстовой строки все пары одинаковых символов.  2. Заменить цифры 5, 2, 0, 4 на буквы X, P, Y, D соответственно. |
| 19 | 1 | 2 | 3 | | 1. Все символы, совпадающие с последним символом строки заменить цифрой «0».  2. Заменить все открывающиеся круглые и фигурные скобки на символ +, закрывающиеся круглые и фигурные скобки на символ - . |
| 20 | 1 | 3 | 2 | | 1. Все символы, совпадающие с первым символом строки заменить цифрой «1».  2. Заменить буквы Ш, Ы, Д, Ц на буквы й, у, т, г соответственно. |

ПРИМЕЧАНИЕ: в таблице обозначены:

1 - оператор ***for***;

2 - оператор ***while***;

3 - оператор ***do-while***.

5. Содержание отчета

5.1. Титульный лист

5.2. Цель работы.

5.3. Задание.

5.4. Текст(ы) программ.

5.5. Полученные результаты.

5.6. Выводы.

6. Контрольные вопросы

6.1. Пояснить логику работы оператора ***if – else*?**Возможны ли вложенные операторы ***if***?

If(условие) {

// Инструкции

}

// Иначе

else {

// Инструкции

}

Возможны.

В качестве дополнительной ветки (else if) или, если потребуется, вложенного if в тело другого If.

6.2. Какой оператор используется для выхода из оператора ***switch***? Для чего используется метка ***default*** в операторе ***switch***?

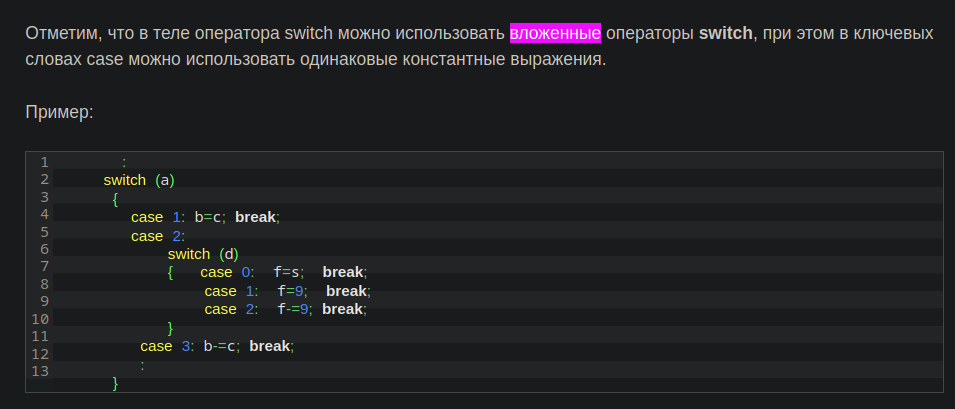
Для выхода из switch используется оператор break.

Метка default задействуется тогда, когда не был осуществлен переход ни по одному из case-ов.

6.3. Реализовать с помощью условной операции функцию ***x = max(a,b)***



6.4. Можно ли использовать выражения для задания констант в операторе ***switch***? Можно ли в теле оператора ***switch*** использовать вложенные операторы ***switch***?



6.5. В каких случаях целесообразно использовать оператор ***goto***?

При большом уровне вложенности.

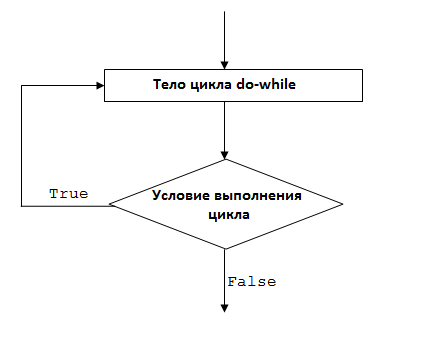
В остальных случаях использование goto нежелательно.

6.6. Написать цикл ***while***, эквивалентный циклу ***for***.

For(int i=0; i < 10; i++) {}

Int i=0; While(i < 10) {i++;}

6.7. Изобразить алгоритм работы цикла ***do-while***.



6.8. В какой из трех конструкций циклов гарантируется хотя бы однократное выполнение тела цикла?

Do while

6.9. Какой оператор прерывает выполнение цикла? Назначение оператора ***continue***.

Прерывает выполнение цикла оператор break.

Continue выполняет переход к следующей итерации.

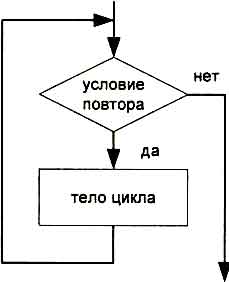
6.10. Как записать цикл с бесконечным числом повторений (бесконечный цикл) для каждого из операторов цикла?

While(true) {} // While(1) {}

For(;;) {}

Do {} while(true)

6.11. Нарисовать блок-схему цикла ***while.***



6.12. Привести пример вложенного цикла.



6.13. Чему равно значение переменной цикла при его нормальном завершении? Прерывании?

При нормальном завершении — ближайшее значение одной из переменных, обращающих условие в false (лучше на примере)

При прерывании — последнее присвоенное значение.

6.14. Можно ли в теле цикла менять переменную цикла?

Можно. Нужно.