

Министерство образования и науки РФ

**НГТУ**

Кафедра Автоматики

**Лабораторная работа № 1**

**Вариант: 6**

**Факультет: АВТФ**

**Группа: АП-819**

**Студент: Зубарев Василий**

**Преподаватель:**

Саблина

Галина

Владимировна

**Дата: 12.02.09**

Новосибирск 2009

**Цель работы:** Изучить процесс написания программы на языке СИ. Ознакомиться с технологией отладки программ. Изучить основные конструкции операторов присваивания и директивы препроцессора.

**Задание:**

A -умножить на B, разделить по модулю C и вычесть D. C и D уменьшить на 1.

B - разделить по модулю 7.

C - сложить с A и B и разделить на 3. A увеличить на 1, B уменьшить на 1.

D - выполнить поразрядное «и» 1 или 5 разрядов B и 3-7 разрядов C, сдвинутых влево на 1 разряд.

**Листинги программ:**

1. Написать программу, выполняющую арифметические и логические операции над целыми переменными *A*, *B*, *C*, *D* по вариантам. Использовать глобальные статические переменные и инициализацию при компиляции. Вывести значения переменных с помощью функции *printf()* в заданном формате.

**Листинг #1**

```
#include <stdio.h>
#include <conio.h> // Я ебал это вендузятское отродье

int A = 12, B = 7, C = 9, D = 17;

void main(int argc, char* argv[])
{
    A = (A * B) % (C--) - (D--);
    B %= 7;
    C = (C + ((A++) + (B--))) / 3;
    D = ((B & 2) | (B & 32)) & ((C & 248) << 1);

    printf("A = %d, B = %d, C = %d, D = %d", A, B, C, D);
    getch();
}
```

2. Модифицировать программу, описав переменные как локальные и задав их значения с помощью прямого присваивания. Сравнить результаты

### Листинг #2

```
#include <stdio.h>
#include <conio.h>

void main(int argc, char* argv[])
{
    int A, B, C, D;
    A = 12; B = 7; C = 9; D = 17;

    A = (A * B) % (C--) - (D--);
    B %= 7;
    C = (C + ((A++) + (B--))) / 3;
    D = ((B & 2) | (B & 32)) & ((C & 248) << 1);

    printf("A = %d, B = %d, C = %d, D = %d", A, B, C, D);
    getch();
}
```

3. Изменить программу, применив для ввода значений переменных функцию *scanf()*. Сравнить результаты.

### Листинг #3

```
#include <stdio.h>
#include <conio.h>

void main(int argc, char* argv[])
{
    int A, B, C, D;

    printf("Enter A, B, C, D\n");
    scanf("%d %d %d %d", &A, &B, &C, &D);

    A = (A * B) % (C--) - (D--);
    B %= 7;
    C = (C + ((A++) + (B--))) / 3;
    D = ((B & 2) | (B & 32)) & ((C & 248) << 1);

    printf("A = %d, B = %d, C = %d, D = %d", A, B, C, D);
    getch();
}
```

4. Описать переменные как автоматические. Вывести их значения до присваивания. Объяснить полученные результаты.

#### Листинг #4

```
#include <stdio.h>
#include <conio.h>

void main(int argc, char* argv[])
{
    auto int A, B, C, D;
    printf("A = %d, B = %d, C = %d, D = %d\n", A, B, C,
D);

    printf("Enter A, B, C, D\n");
    scanf("%d %d %d %d", &A, &B, &C, &D);

    A = (A * B) % (C--) - (D--);
    B %= 7;
    C = (C + ((A++) + (B--))) / 3;
    D = ((B & 2) | (B & 32)) & ((C & 248) << 1);

    printf("A = %d, B = %d, C = %d, D = %d", A, B, C, D);
    getch();
}
```

В первом printf после объявления переменных на экран выводится «программный мусор» - содержимое кусков памяти, которые были отведены под переменные.

5. Повторить, разместив все описания в отдельном файле.

Создадим файл vars.c в котором содержится строка `int A, B, C, D;`

#### Листинг #5

```
#include <stdio.h>
#include <conio.h>
#include "vars.c"

void main(int argc, char* argv[])
{
    A = (A * B) % (C--) - (D--);
    B %= 7;
    C = (C + ((A++) + (B--))) / 3;
    D = ((B & 2) | (B & 32)) & ((C & 248) << 1);

    printf("A = %d, B = %d, C = %d, D = %d", A, B, C, D);
    getch();
}
```

6. Повторить, используя вместо переменных *A*, *B*, *C*, *D* статический массив.

### Листинг #6

```
#include <stdio.h>
#include <conio.h>

void main(int argc, char* argv[])
{
    static int arr[] = {12, 7, 9, 17};

    // Индексы в массиве: A - 0, B - 1, C - 2, D - 3
    arr[0] = (arr[0] * arr[1]) % (arr[2]--) - (arr[3]--);
    arr[1] %= 7;
    arr[2] = (arr[2] + ((arr[0]++) + (arr[1]--))) / 3;
    arr[3] = ((arr[1] & 2) | (arr[1] & 32)) & ((arr[2] &
248) << 1);
    printf("A = %d, B = %d, C = %d, D = %d", arr[0],
arr[1], arr[2], arr[3]);
}
```

### Пример вывода данных программ:

Исходные данные:

A = 12, B = 7, C = 9, D = 17

Вывод:

A = -13, B = -1, C = -2, D = 32

**Вывод:** Мы познакомились с языком программирования C, написали простые программы для работы с целыми числами и отладили ее. Освоили директивы препроцессора и типы переменных. Полученные результаты не зависят от метода объявления переменных, как и должно было быть.