

Repeat & Plot

C++ Basic - lesson 1

1 Простой калькулятор целых чисел

[lesson_01/task_1_calc.cpp]

Пользователь вводит два целых числа. Вывести на экран сумму, разность, частное и остаток от деления первого числа на второе.

Пример ввода / вывода на консоль:

```
Enter A: 17
```

```
Enter B: 5
```

```
17 + 5 = 22
```

```
17 - 5 = 12
```

```
17 * 5 = 85
```

```
17 div 5 = 3
```

```
17 mod 5 = 2
```

2 Обмен значений переменных

[lesson_01/task_2_swap.cpp]

Пользователь вводит два числа. Одно присваивается одной переменной, а второе другой. Необходимо поменять значения переменных так, чтобы значение первой оказалось во второй, а второй - в первой.

3 Минимальное из трех

[lesson_01/task_3_min.cpp]

Пользователь вводит три различных числа. Вывести минимальное из них.

4 Таблица

[lesson_01/task_4_table.cpp]

Вывести на экран таблицу для построения параболы по точкам. В первом столбце должны располагаться значения x в диапазоне от a до b . Во втором столбце - соответствующее значение $y = x^2$. Значения a , b и N пользователь вводит с клавиатуры, где N — количество строк в таблице (т. е. точек разбиения).

Прим.: используйте тип `double` вместо `int`.

5 Графики в gnuplot

При запуске программы перенаправьте поток вывода с консоли в файл **table.dat**:

```
./a.out > table.dat
```

После этого запустите программу для построения графиков **gnuplot**:

```
comp@comp-H61H-G11:~$ gnuplot
```

Чтобы построить график по точкам из файла **table.dat** выполните команду **plot** в среде **gnuplot**, в качестве аргумента укажите название файла с точками для построения графика:

```
gnuplot> plot "table.dat"
```

Попробуйте следующий набор команд, чтобы оценить удобство программы:

```
gnuplot> set title "Название графика"
```

```
gnuplot> set ylabel "Подпись оси ординат"
```

```
gnuplot> set xlabel "Подпись оси абсцисс"
```

```
gnuplot> plot "table.dat" with lines
```

Внешний вид графика определяется набором команд, подобных тем, что представлены выше. Каждый раз вводить команды долго. Для удобства можно написать все команды один раз и сохранить в текстовый файл. Текстовые файлы с набором команд называются скриптами.

Сохраните следующий скрипт в файл с именем **build.plt**:

```
set ylabel "y"
```

Grade_10 (free use), © Igonin Semen, Saratov

```
set xlabel "x"
```

```
set title "Parabola"
```

```
plot "table.dat" with lines
```

```
pause -1
```

Grade_10 (free use), © Igonin Semen, Saratov

Теперь построить график можно двумя способами.

Способ 1:

1. Запустите программу **gnuplot**:

```
comp@comp-H61H-G11:~$ gnuplot
```

2. Загрузите ранее подготовленный скрипт:

```
gnuplot> plot "build.plt"
```

Способ 2:

1. Запустите программу **gnuplot**, передав через пробел в качестве аргумента название скрипта `build.plt`:

```
comp@comp-H61H-G11:~$ gnuplot build.plt
```