

Сначала мы проверяем наши флаги. Если есть «--fromfile» или «--tofile», мы это запоминаем в переменных flag1, flag2.

```
bool flag1 = false, flag2 = false;
string ofname;
if (argc >= 3) {
    if (strcmp(argv[1], "--fromfile") == 0) flag1 = true;
    if (strcmp(argv[1], "--tofile") == 0) {
        flag2 = true;
        ofname = argv[2];
    }
}
if (argc == 5) {
    if (strcmp(argv[3], "--fromfile") == 0) flag1 = true;
    if (strcmp(argv[3], "--tofile") == 0) {
        flag2 = true;
        ofname = argv[4];
    }
}
```

Потом мы просто делаем считывание. Моя программа рассчитана на ввод некоторого числа данных, то есть можно обработать сразу несколько людей. Поэтому мы пробегаемся циклом и не только считываем данные, но и их обрабатываем.

Потом мы вызываем соответственно функции вывода output1 и output2. Первая выводит ответ в консоль, вторая в заданный пользователем файл. Их реализация в funcs.h.

```
void output1(string name, string s, int y, int m, int d) {
    if (name.compare("MIEM") == 0) {
        GeneratorMIEM a;
        cout << a.generate(s, y, m, d) << endl;
    }
    else {
        GeneratorMGUUU a;
        cout << a.generate(s, y, m, d) << endl;
    }
}

void output2(ofstream* file, string name, string s, int y, int m, int d) {
    if (name.compare("MIEM") == 0) {
        GeneratorMIEM a;
        *file << a.generate(s, y, m, d) << endl;
    }
    else {
        GeneratorMGUUU a;
        *file << a.generate(s, y, m, d) << endl;
    }
}
```

Реализация классов тоже в funcs.h. Там я создал базовый абстрактный класс Base, где создал виртуальные методы для будущих классов. Далее создаю 2 класса для 2 университетов соответственно, наследуемые от класса Base. Переопределяю классы. Для более лаконичного подсчёта суммы цифр и билете (что нужно для определения последней цифры в билете) создана функция number().

```
int number(string s) {
    int res = 0;
    for (int i = 0; i < s.length(); i++) {
        res += ((int)s[i] - (int)'0') * (i + 1);
    }return res;
}
```

Также для форматирования случайных чисел в билетах была создана функция `help()`б которая на вход получает не только само случайное число, но и кол-во необходимых чисел. Возвращает просто необходимую строку из нулей.

```
string help(int n, int a) {
    string res = "";
    while (a > 0) {
        a /= 10;
        n--;
    }for (int i = 0; i < n; i++) {
        res += "0";
    }return res;
}
```

Makefile реализован довольно простенько. Одной командой (просто `make`) мы создаём экзешник. Команды `clean` и `distclean` работают, как и требовалось.

В итоге мы получаем работающую программу. Примеры ввода и вывода для работы с файлами можно найти в текстовых файлах `in.txt` и `out.txt`.