

ФИО: Ребдев Павел Александрович  
Группа: 5130904/30008  
Лабораторная работа: «Операторы цикла»

### Постановка задачи

Разработать детальные требования и тест план для следующей задачи:

**Целые степени 2.** По данному числу N распечатайте все целые степени двойки, не превосходящие N, в порядке возрастания. Операцией возведения в степень пользоваться нельзя!

### Детальные требования

1. Число N должно быть задано корректно:
  - 1.1. N — число. Если N не является числом, то выводится сообщение об ошибке и программа завершается
2. Вывод всех степеней двойки до N:
  - 2.1. Если  $N \leq 0$ , то ничего не выводится
  - 2.2. Если  $N = 1$ , то выводится 1
  - 2.3. Если  $(N \% 2) == 1$ , то выводятся степени двойки, принадлежащие промежутку  $[1; (N/2)*2]$
  - 2.4. Если  $(N \% 2) == 0$ , то выводятся степени двойки, принадлежащие промежутку  $[1; N]$

### Тест-план

Проверка детальных требований с помощью тест-плана:

| #   | Описание   | Результат                                     |
|-----|--|---|
| 1.1 | N — число. Если N не является числом, то выводится сообщение об ошибке и программа завершается | Input:<br>notNum<br>Expected:<br>Input error! |
|     |  | Input:<br>g124<br>Expected:<br>Input error!   |
| 2.1 | Если $N \leq 0$ , то ничего не выводится   | Input:<br>- 12<br>Expected:                   |
| 2.2 | Если $N = 1$ , то выводится 1  | Input:<br>1<br>Expected:<br>1                 |
| 2.3 | Если $(N \% 2) == 1$ , то выводятся степени двойки, принадлежащие промежутку $[1; (N/2)*2]$    | Input:<br>11<br>Expected:<br>1<br>2<br>4<br>8 |
| 2.4 | Если $(N \% 2) == 0$ , то выводятся степени двойки, принадлежащие промежутку $[1; N]$          | Input:<br>12<br>Expected:<br>1<br>2<br>4<br>8 |

### Исходные тексты программы

Файлы с исходными текстами лабораторной работы (полагаем <R00T> для папки в котором располагаются исходные тексты):

**./<ROOT>/main.cpp**

```
#include <iostream>
#include <limits>

int main()
{
    long long int limit = 0;
    std::cin >> limit;

    if(!std::cin)
    {
        std::cerr << "Input error!\n";
        return 1;
    }
    long long int num = 1;

    while ((num <= (std::numeric_limits< long long int >::max()/2))
    && (num <= limit))
    {
        std::cout << num << '\n';
        num *= 2;
    }
    return 0;
}
```