

Начнём сразу с задания. Нам дают вычислять функциональный ряд:

$$f(x) = \frac{(U*x)^1}{1!} - \frac{(U*x)^3}{3!} + \frac{(U*x)^5}{5!} - \frac{(U*x)^7}{7!} \dots$$

Чтобы посчитать значение функции, нам надо понять как его считать. Наш функциональный ряд представляет собой большую сумму, состоящую из бесконечного числа слагаемых.

$$f(x) = \frac{(U*x)^1}{1!} - \frac{(U*x)^3}{3!} + \frac{(U*x)^5}{5!} - \frac{(U*x)^7}{7!} \dots$$

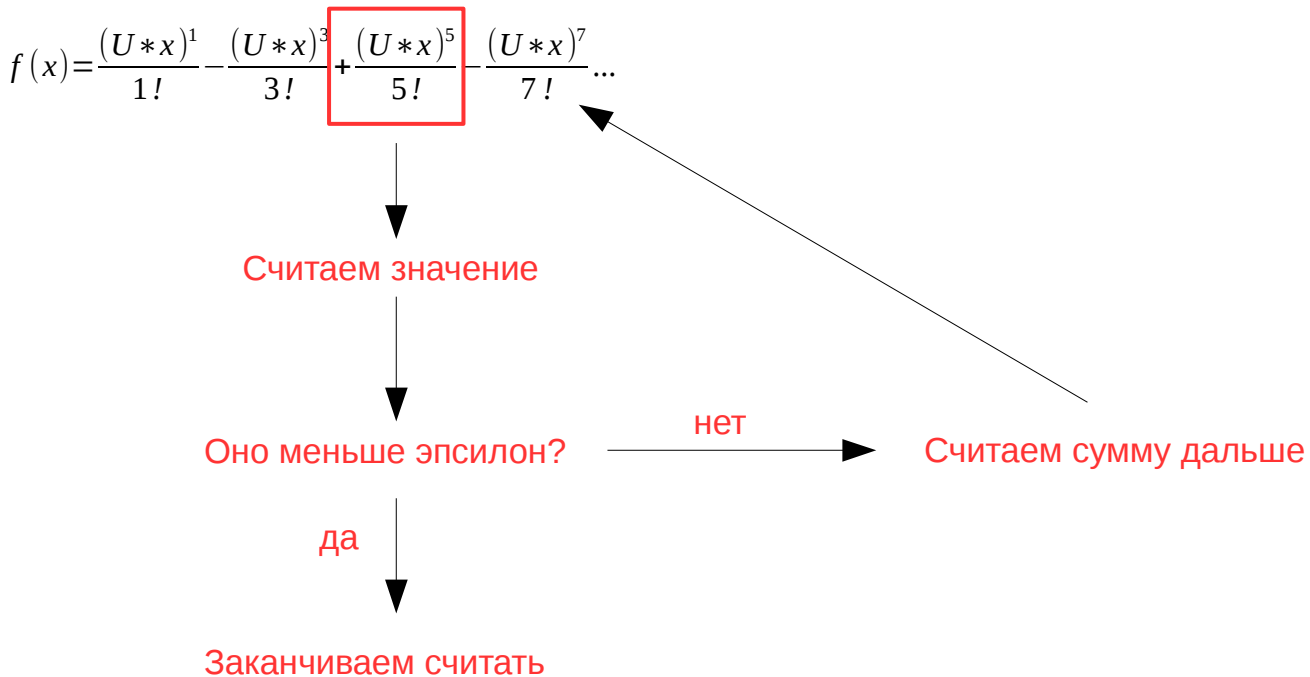
← слагаемое

← их много

Мы не имеем ни времени, ни ресурсов считать сумму бесконечно и поэтому нам надо понять когда остановится чтобы получить значение функции с определённой точностью.

Для этого мы берём какое-нить малое число (я буду звать его эпсилон). Потом считаем каждый слагаемое и смотрим, чтобы оно было больше или равно нашего эпсилон. Если очередное слагаемое меньше эпсилон, значит мы посчитали значение функции с достаточной точностью.

$$\varepsilon = 10^{-4}$$



Теперь у нас есть простенький алгоритм, а следовательно мы уже можем перейти к программированию. Но прежде давайте упростим наш функциональный ряд, чтобы не париться при написании цикла.

Как говорилось ранее, наш функциональный ряд это просто бесконечная сумма слагаемых, которые постоянно меняют свой знак. Так что давайте приведём нашу функцию в более рентабельный для программиста вид:

$$f(x) = \frac{(U*x)^1}{1!} - \frac{(U*x)^3}{3!} + \frac{(U*x)^5}{5!} - \frac{(U*x)^7}{7!} \dots \longrightarrow f(x) = \sum_{i=1}^{\infty} \frac{(V*x)^{(i+(i-1))}}{(i+(i-1))!} * (-1)^{(i+1)}$$

Задаётся вполне резонный вопрос о том, как я к этой формуле пришёл. Не важно, главное что она работает. Сами подставьте вместо  $i$  какое нибудь число и получите  $i$ -ый элемент суммы.

Обратите внимание на то, как я реализовал смену знака домножая на  $-1$  в какой-то степени и то, как я реализовывал подсчёт степени и знаменателя ( $1, 3, 5 \dots$ ).

А перевод функционального ряда к сумме выполнить очень легко, советую самим этот трюк повторить.

И переходим к наброску нашей программы:

```
double v = 6; // номер варианта (поставь что угодно если не помнишь)
double epsilon = pow(10,-4); // та самая маленькая цифра
double sum = 0; // сюда будем накапливать сумму
int i = 1; // номер слагаемого
double x = 1; // аргумент функции

// следующая строчка вынесет мозг тем, кто программирует два с половиной дня, а по
// сути это просто бесконечный цикл. While (условие) и тело. Если в условие впихнуть
// истину при любом раскладе, то и считать он будет вечность.
while (true) {
    // подсчитываем наше слагаемое
    double element = pow(-1, i+1) * (pow(v*x, i+(i-1)) / fact(i+(i-1)));
    // несложно проверить (если вы не Повстаной), что предыдущая строчка совпадает
    // с формулой.

    // Теперь нам надо сравнить наше слагаемое, которое мы только что получили
    // с эпсилон, чтобы понять, а стоит ли нам считать дальше или мы уже получили
    // необходимую точность.
    if (abs(element) < epsilon) break;
    // abs(element) это модуль нашего слагаемого. Если он меньше epsilon, то break
    // break — это послать цикл к чертям и приостановить его выполнение

    // Но если слагаемое не меньше эпсилон, то
    else {
        sum += element; // прибавляем к сумме слагаемое
        i++; // переходим к следующему номеру элемента
    }
}
```

Но если вы спишете то, что я написал выше, то ваша программа работать не будет. Причиной тому, отсутствие функции для подсчёта факториала (видимо те, кто писал `cmath` никогда факториалы не считали), которую (к вашему сожалению) нам придётся написать самому.

В общем, мне не стоит рассказывать как считать факториал и как привести программу к рабочему виду, так как это довольно тривиальная задача. Я упростил свой изначальный код, сделал его более простым и коротким и опубликовал его здесь:

<https://github.com/VadimGush/GushMath/blob/master/project1/example/main.cpp>

Там опубликован код уже работающей программы и в ней же, функция для вычисления факториала.



Это была первая часть. Поплавский это задание решил ещё усложнить так, чтобы вы парились с вычислением значений функции на отрезке (“и целого мира мало”).

Так что пока переварите написанное выше.  
Выпейте чаю и съешьте торт  
(это приказ)

