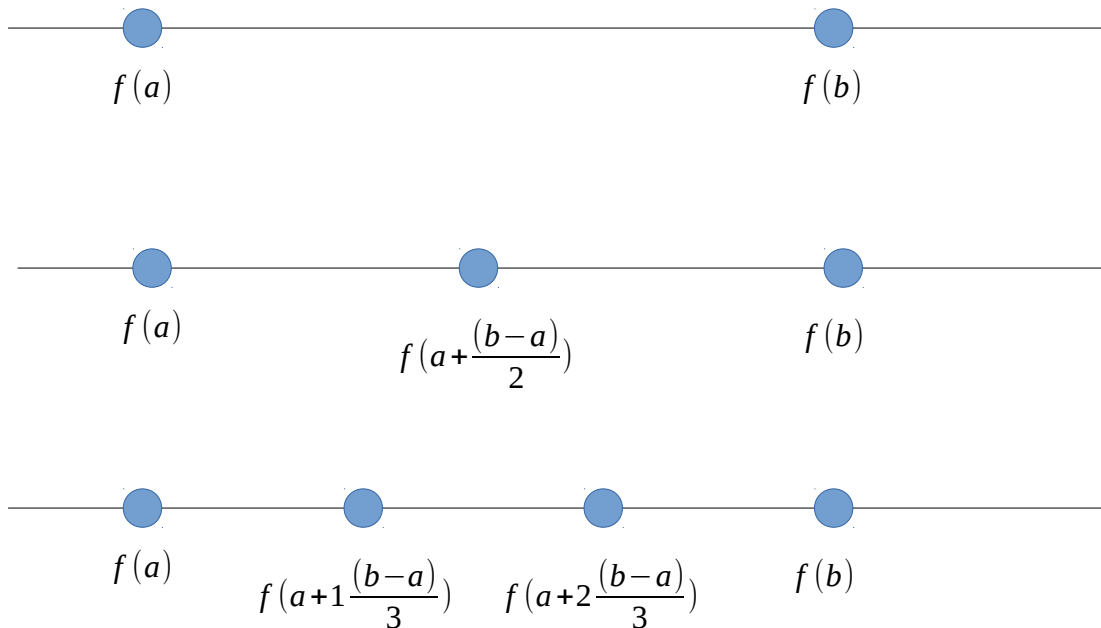


Теперь нам стоит переписать программу в более удобный вид. Вынесем код, который подсчитывает функциональный ряд в отдельную функцию и продолжим выполнять задание.

Ссылка на новый файл:

<https://github.com/VadimGush/GushMath/blob/master/project1/example/main2.cpp>

Главный смысл заключается в том, что нам даётся какой-то отрезок на прямой. На этом отрезке мы должны посчитать значения функции:



То есть если нам даётся отрезок  $a=5$  и  $b=8$ , разделённый на три части, то мы должны вывести на экран значения  $f(5)$ ,  $f(6)$ ,  $f(7)$ ,  $f(8)$ . То есть по сути мы всегда выводим на экран количество элементов на единицу больше, чем количество заданных частей.

Скажем что количество сегментов это  $S$ , тогда:

$$\Delta = \frac{(b-a)}{S} \quad - \text{ что есть расстояние между точками на отрезке}$$

Реализуем это в коде:

```
float a = 2; // начало отрезка
float b = 5; // конец отрезка
int segments = 3; // количество сегментов
float delta = (b-a) / segments; // расстояние между точками

for (int i = 0, i <= segments; i++) {
    cout << sum(a + delta * i) << endl; // выводим значение функции в конкретной точке
}
```

Функция `sum()` это та самая функция, в которую мы вынесли вычисление функционального ряда. Теперь код стал более простым и понятным.

Теперь всё что вам осталось сделать, это чтобы программа первой строкой выводила точки на отрезках, второй строкой значения функции в этих точках, а третьей количество слагаемых в сумме (как это реализовать придумайте сами). Отрисовку таблиц я надеюсь объяснять не стоит, так как мы проходили это на первом курсе.

**Внимание:** не берите точки меньше -8 и больше 8, потому что из-за ошибок округления погрешность станет настолько большой, что значения функции будут просто нереалистичными и заоблачными.

Вывод программы должен быть примерно такой. Как мы видим пользователь задаёт количество сегментов, точки `a` и `b`, а в ответ получает таблицу, где первой строкой точки на отрезке, второй строкой значения функции, а третьей строкой количество слагаемых при вычислении суммы.

```
Write epsilon (0 for default value): 0
Count of segments (0 for default value): 3
Write a: 2
Write b: 5
-----
2          3          4          5
-0.536518  -0.750909  -0.905589  -0.987909
19         27         36         44
-----
darina@darina-lin:~/Documents/GushMath/project1$
```