

Лабораторная работа №1
Табулирование функций.

Изучение среды программирования и основных операторов базового языка. Разработка алгоритма, разработка, отладка и выполнение программы табулирования функций.

Краткие теоретические сведения:

Алгоритм – совокупность точно заданных правил решения некоторого класса задач или набор инструкций, описывающих порядок действий исполнителя для решения определённой задачи.

Шаг алгоритма - каждое отдельное действие алгоритма.

Алгоритмизация – это процесс построения алгоритма решения задачи, результатом которого является выделение этапов процесса обработки данных, формальная запись содержания этих этапов и определение порядка их выполнения.

Алгоритм табулирования функции:

1. Задаем начальное значение x , конечное значение x и шаг.
2. Итерируемся по значениям x в заданном диапазоне, используя заданный шаг.
3. Для каждого значения x , находим значение функции $f(x)$.
4. Сохраняем полученные пары значений $(x, f(x))$.
5. Повторяем шаги 2-4 для всех значений x в диапазоне.
6. Выводим полученные пары значений $(x, f(x))$ их можно записать в массив или файл, если нужно.

Задание 1. Изучите содержимое программы. Постройте блок-схему алгоритма.

```
#include <iostream>
```

```
int main() {
```

```
    double start, end, step;
```

```
    std::cout << "Введите начальное значение x: ";
```

```
    std::cin >> start; // Задаем исходное значение переменной
```

```
    std::cout << "Введите конечное значение x: ";
```

```
    std::cin >> end; // Задаем конечно значение переменной
```

```
    std::cout << "Введите шаг: ";
```

```
    std::cin >> step; // Задаем шаг, с которым с которым будет изменяться
```

аргумент

```
    for (double x = start; x <= end; x += step) {
```

```
        double y = sin(x); // функция  $y = \sin(x)$ 
```

```
        std::cout << x << " | " << y << std::endl; // Выводим значения
        функции для каждого аргумента (x | y)
    }
    return 0;
}
```

Задание 2. Проведите отладку программы.

Процесс отладки включает в себя:

1. Воспроизведение ошибки разработчиков, чтобы понять, какая часть программы не работает правильно
2. Изменение кода разработчиком с целью устранения ошибки
3. Запуск отладчика - инструмента, который позволяет программисту выполнить код по шагам
4. Изучение программистом логов, которые генерируются во время разработки программы и могут помочь в обнаружении ошибок
5. Повторное тестирование для проверки работоспособности программы после исправления ошибки