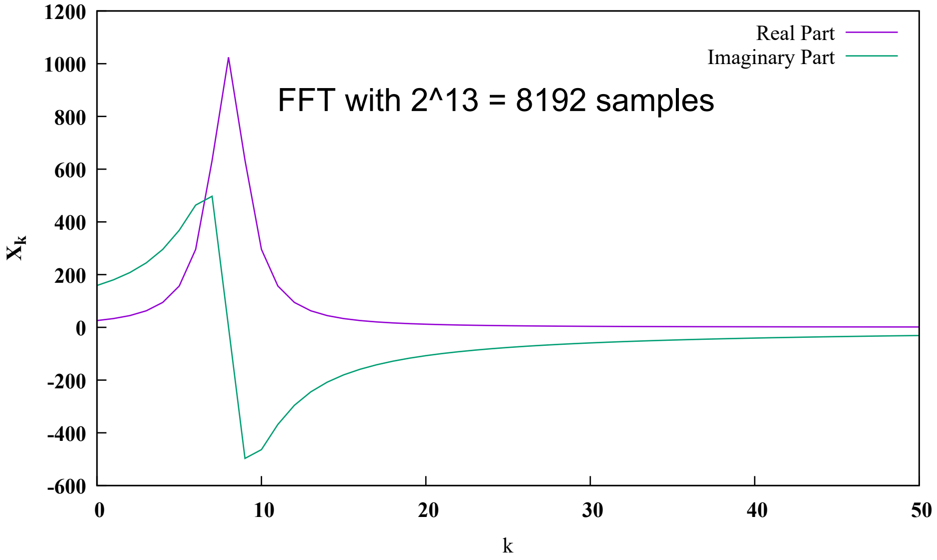
Rajat Kumar Mandal - 226121014

```cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <complex>

//Types used in the problem
typedef std::vector<std::complex<double>> Complexlist;

//Constant expressions appearing in the problem
constexpr double pi = 3.14159265359;
constexpr std::complex<double> i(0, 1);

//This is the function to be taken DFT of
std::complex<double> function(double t) {
    return exp(2.0 * pi * i * t - t);
}

//Helper function to discretize the given function
Complexlist discretizer(std::complex<double>(*function)(double t), double
  starting_point, double ending_point, size_t STEPS) {
    Complexlist result(STEPS);
    for (size_t i = 0; i < STEPS; i++) {
        result[i] = function(starting_point + i * (ending_point -
          starting_point) / (STEPS - 1));
    }

    return result;
}

//FFT algorithm
Complexlist FFT(Complexlist P) {
    size_t n = P.size();

    if (n == 1) {
        return P;
    }

    //n roots of unity
    std::complex<double> w = std::exp(-2.0 * pi * i / (double)n);

    Complexlist Pe, Po;

    //Divide into odd and even polynomials
    for (size_t j = 0; j < n / 2; j++) {
        Pe.push_back(P[2 * j]);
        Po.push_back(P[2 * j + 1]);
    }

    //invoke recuesion
    Complexlist ye = FFT(Pe), yo = FFT(Po), y(n);

    //tie up everything to complete
    for (size_t j = 0; j < n / 2; j++) {
        y[j] = ye[j] + pow(w, j) * yo[j];
        y[j + n / 2] = ye[j] - pow(w, j) * yo[j];
    }

    return y;
}

int main() {
    //Discretize given function in 2048 steps
    Complexlist coefficients = discretizer(function, 0, 8, 2048);

    //Calculate the FFT
    Complexlist values = FFT(coefficients);

    std::ofstream outfile;  //file handle to save the results in a file
    outfile.open("./output/FFT.txt", std::ios::out | std::ios::trunc);

    for (auto& value : values) {  //Loop through the array to store the values
        outfile << value.real() << "\t" << value.imag() << std::endl; // write to the output file
    }
    outfile.close();    //when done, close the file.

    return 0;
}
```



FFT with 2^13 = 8192 samples
Real Part
Imaginary Part
1200
1000
800
600
400
200
0
-200
-400
-600
X_k
0
10
20
30
40
50
k