```
#include <iostream>
#include <fstream>
#include <string>
#include <cstring> // Required for strncpy

using namespace std;

const int RECORD_SIZE = sizeof(int) + 50; // Assuming an integer ID and a string of max length 50
const char* FILENAME = "records.dat";

struct Record {
    int id; // Unique identifier
    char name[50]; // Name of the record

    Record() : id(0) {
        name[0] = '\0'; // Initialize name to an empty string
    }
};

// Function to insert a record at a specific position
void insertRecord(int position, const Record& record) {
    fstream file(FILENAME, ios::in | ios::out | ios::binary);
    if (!file) {
        // If the file doesn't exist, create it
        file.open(FILENAME, ios::out | ios::binary);
        file.close();
        file.open(FILENAME, ios::in | ios::out | ios::binary);
    }

    // Move to the position in the file
    file.seekp(position * sizeof(Record));
    file.write(reinterpret_cast<const char*>(&record), sizeof(Record));
    file.close();
    cout << "Record inserted at position " << position << endl;
}

// Function to delete a record by setting its ID to -1
void deleteRecord(int position) {
    fstream file(FILENAME, ios::in | ios::out | ios::binary);
    if (!file) {
        cout << "File does not exist." << endl;
        return;
    }
    Record record;
    file.seekg(position * sizeof(Record));
    file.read(reinterpret_cast<char*>(&record), sizeof(Record));

    if (record.id != 0) { // Check if the record exists
        record.id = -1; // Mark the record as deleted
        file.seekp(position * sizeof(Record));
        file.write(reinterpret_cast<const char*>(&record), sizeof(Record));
        cout << "Record deleted at position " << position << endl;
    } else {
        cout << "No record found at position " << position << endl;
    }

    file.close();
}

// Function to display all records
void displayRecords() {
    fstream file(FILENAME, ios::in | ios::binary);
    if (!file) {
        cout << "File does not exist." << endl;
        return;
    }

    Record record;
    int position = 0;
    cout << "\nRecords in the file:" << endl;
    while (file.read(reinterpret_cast<char*>(&record), sizeof(Record))) {
        if (record.id != -1) { // Only display valid records
            cout << "Position: " << position << ", ID: " << record.id << ", Name: " << record.name << endl;
        }
        position++;
    }

    file.close();
}

int main() {
    Record record1, record2, record3;

    record1.id = 1;
    strncpy(record1.name, "Rahul", sizeof(record1.name));
    record1.name[sizeof(record1.name) - 1] = '\0';
    record2.id = 2;
    strncpy(record2.name, "Priya", sizeof(record2.name));
    record2.name[sizeof(record2.name) - 1] = '\0';

    record3.id = 3;
    strncpy(record3.name, "Amit", sizeof(record3.name));
    record3.name[sizeof(record3.name) - 1] = '\0';

    insertRecord(0, record1);
    insertRecord(1, record2);
    insertRecord(2, record3);

    displayRecords();

    deleteRecord(1); // Delete Priya

    displayRecords();

    return 0;
}
```