```cpp
#include "hash_table.h"
#include <iostream>
#include <fstream>
#include <chrono>
#include <random>

using namespace std;

void generate_file(ofstream& of, int count_lines);



void main() {
	srand(time(0));
	setlocale(LC_ALL, "rus");

	hash_table* table = new hash_table(0);

	string path_in = "input.txt";
	string path_out = "output.txt";
	int count_lines = 100000;
	int command_number = 0;
	ofstream wf("input.dat", ios::out | ios::binary);
	generate_file(wf,count_lines);

	do
	{
		patient* tmp = new patient();
		std::cout << "Enter command code: "
			<< "\n0. Enter new table"
			<< "\n1. Add antoher patient"
			<< "\n2. Delete patient from file"
			<< "\n3. Find patient"
			<< "\n4. Display table"
			<< "\n5. Clear table"
			<< "\nexit : -1"
			<< "\nYour input: ";
		std::cin >> command_number;

		switch (command_number)
		{
		case 0:
		{
			if (!table->isEmpty()) table->clear();
			cout << "\nLocate the file: ";
			cin >> path_in;
			ifstream fin(path_in, ios::binary | ios::in);
			table->set_file_path(path_in);
			int index = 0;
			int prev_card_number = 0;
			bool get_the_numb = true;
			if (fin.is_open()) {
				while (!fin.eof()) {
					fin.read((char*)&(*tmp), sizeof(patient));

					if (prev_card_number == tmp->card_number)
continue;
					prev_card_number = tmp->card_number;
```

```
                                    bool res = table->add(*tmp,
fin.tellg()/sizeof(patient) - 1 );
                                    if (res)
                                    {
                                          index++;
                                          if (index >= (count_lines / 2) and
get_the_numb) {
                                                get_the_numb = false;
                                                cout << tmp->card_number << " "
<< tmp->illness_code << " " << tmp->doctor_surname  << " " << index-1 << "\n";
                                          }
                                    }
                              }
                        }
                        fin.close();
                        break;
                  }
                  case 1:
                  {
                        cout << "\nEnter patient's info: ";
                        cin >> tmp->card_number >> tmp->illness_code >> tmp-
>doctor_surname;
                        tmp->index = table->get_current_size();
                        ofstream fin(table->get_file_path(), ios::binary |
ios::out | ios::app);
                        fin.write((char*)&(*tmp), sizeof(patient));
                        table->add(*tmp, table->get_current_size());
                        break;
                  }
                  case 2:
                  {
                        int card_number;
                        cout << "\nEnter patient's card number: ";
                        cin >> card_number;


                        table->delete_field(card_number);
                        table->display_all_file(table->get_file_path());
                        table->clear();
                        ifstream fin(table->get_file_path() , ios::binary |
ios::in);
                        int index = 0;
                        int prev_card_number = 0;
                        if (fin.is_open()) {
                              while (!fin.eof()) {
                                    fin.read((char*)&tmp, sizeof(patient));
                                    if (prev_card_number == tmp->card_number)
continue;
                                    prev_card_number = tmp->card_number;
                                    if (table->add(*tmp, index)) index++;
                              }
                        }
                        fin.close();

                        break;
                  }
                  case 3:
                  {
```

```
                    int card_number;
                    cout << "\nEnter patient's card number: ";
                    cin >> card_number;
                    int start_time = clock();
                    auto start = chrono::high_resolution_clock::now();
                    tmp = table->get_patient(card_number);
                    if (tmp->card_number == -1)
                    {
                          cout << "He is not our patient\n";
                          continue;
                    }
                    ifstream fdirect(table->get_file_path(), ios::in |
ios::binary);
                    fdirect.seekg((int)(tmp->index)*sizeof(patient), ios::beg);
                    fdirect.read((char*)&(*tmp), sizeof(patient));
                    fdirect.close();
                    auto elapsed = chrono::high_resolution_clock::now() -
start;
                    long long microseconds =
std::chrono::duration_cast<std::chrono::microseconds>(elapsed).count();
                    cout << tmp->card_number << " " << tmp->illness_code << "
" << tmp->doctor_surname << " " << tmp->index << " " << " //time - " <<
microseconds << "\n";

                    break;
              }
              case 4:
              {
                    table->display_all();
                    break;
              }
              case 5:
              {
                    table->clear();
              }

              case -1:
                    return;

              }

              std::cout << std::endl;
        } while (command_number != -1);


}
```