Telco Data Check & Analyze

- Write a C++ program to perform some queries on a telco data (comming from stdin) with the following format:
- The first block of data consists of lines (terminated by a line containing #), each line (number of lines can be up to 100000) is under the form:

  call <from_number> <to_number> <date> <from_time> <end_time>

  - which is a call from the phone number <from_number> to a phone number <to_number> on <date>, and starting at time-point <from_time>, terminating at time-point <end_time>
  - <from_number> and <to_number> are string of 10 characters (a phone number is correct if it contains only digits 0,1,...,9, otherwise, the phone number is incorrect)
  - <date> is under the form YYYY-MM-DD (for example 2022-10-21)
  - <from_time> and <to_time> are under the form hh:mm:ss (for example, 10:07:23)
- The second block consists of queries (terminated by a line containing #), each query in a line (number of lines can be up to 100000) and belongs to one of the following types:
  - ?check_phone_number: print to stdout (in a new line) value 1 if no phone number is incorrect
  - ?number_calls_from <phone_number>: print to stdout (in a new line) the number of times a call is made from <phone_number>
  - ?number_total_calls: print to stdout (in a new line) the total number of calls of the data
  - ?count_time_calls_from <phone_number>: print to stdout (in a new line) the total time duration (in seconds) the calls are made from <phone_number>

Implementation

```
#include <bits/stdc++.h>
using namespace std;
bool checkPhone (string s){
    if (s.length() != 10) return false;
    for (int i=0; i<s.length(); i++)
        if (!(s[i]>='0' && s[i]<='9')) return false;
    return true;
}
int countTime (string ftime, string etime){
    int startTime = 3600*((ftime[0]-'0')*10 + ftime[1]-'0') + 60*((ftime[3]-'0')*10 + ftime[4]-'0') +
                ((ftime[6]-'0')*10 + ftime[7]-'0');
    int endTime = 3600*((etime[0]-'0')*10 + etime[1]-'0') + 60*((etime[3]-'0')*10 + etime[4]-'0') +
                ((etime[6]-'0')*10 + etime[7]-'0');
    return endTime - startTime;
}
map <string,int> numberCalls, timeCall;
```

Telco Data Check & Analyze - Hint

- Use mapping data structure

  map<string, int>

  to map a phone number to some information

Telco Data Check & Analyze

- Example

stdin	stdout
call 0912345678 0132465789 2022-07-12 10:30:23 10:32:00 call 0912345678 0945324545 2022-07-13 11:30:10 11:35:11 call 0132465789 0945324545 2022-07-13 11:30:23 11:32:23 call 0945324545 0912345678 2022-07-13 07:30:23 07:48:30 # ?check_phone_number ?number_calls_from 0912345678 ?number_total_calls ?count_time_calls_from 0912345678 ?count_time_calls_from 0132465789 #	1 2 4 398 120

Implementation

```
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(NULL);
    cout.tie(NULL);
    string type;
    int totalCalls = 0;
    int incorrectPhone = 0;
    do {
        cin >> type;
        if (type == "#") continue;
        ++totalCalls;
        string fnum, tnum, date, ftime, etime;
        cin >> fnum >> tnum >> date >> ftime >> etime;
        if (!checkPhone(fnum) || !checkPhone(tnum)) ++incorrectPhone;
        numberCalls[fnum]++;
        timeCall[fnum] += countTime(ftime, etime);
    }  while (type!="#");
```

Implementation

```
do {
        cin >> type;
        if (type == "#") continue;
        if (type == "?check_phone_number") {
            if (incorrectPhone == 0) cout << 1 << endl;  else cout << 0 << endl;
        } else if (type == "?number_calls_from") {
            string phone;   cin >> phone;
            cout << numberCalls[phone] << endl;
        }else if (type == "?number_total_calls")
            cout << totalCalls << endl;
        else if (type == "?count_time_calls_from") {
            string phone;  cin >> phone;
            cout << timeCall[phone] << endl;
        }
    }while (type!="#");
    return 0;
}
```