Contacts

Attributes : 1 - full name, 2 - phone number, 3 - e-mail, 4 - address, 5 -job and 6 - relationship of the contact to the user. All the attributes are written as strings. Max number of the contacts is set to be 20(L) and characters inside every string is set to be 100(N).

int count = 10 : in the .c file there is 10 test cases prepared beforehand, that is why count is set to 10, so new contacts can be added(deleted) as the ones after 10th(0~9).

int main() : Only includes heading design, menu and switch case to control the functions. I used char for scanning menu's operation numbers to avoid infinite loop that occurs when %d is used and while (getchar() != '\n'); at the end to make the menu avoid from being printed twice.

void add_contact() : The function adds a contact by asking the user to enter each required attribute. As each attribute being entered it calls matching Boolean function to check whether the input is meeting the requirements or not. while (error_full_name(full_name[count]) == false) : if the entered full name has characters besides letters and an underscore in the middle, it prints an error message and asks the user to make the input again until it is true (meets all the requirements). &full_name[count] : the entered string is being written to the 11th(full_name[10]) slot of the existing array of string full_name. The same things are repeated for each attribute. At the end of the function count increments to give access to the next contact which will be included.

bool error_full_name(char full_name[L][N]) :

```
for (int i = 0; i < 1; i++)-> set to iterate only once, because it
has to check only one string at a time

     {
          for (int j = 0; j < strlen(full_name[i]); j++)-> checks
every character of the string, that is why ends when the string
reaches its length
          {
               if (isalpha(full_name[0][0])) {
                    if (full_name[i][j] == '_' && j !=
strlen(full_name[i]) - 1) {
                         ++count_underscore;   ->   if   the   first
character of the string is a letter and one character is an underscore
and also that underscore is not the last character of the string
count_underscore which initially was 0 increments by 1.


     if (count_underscore == 1) {
        for (int i = 0; i < 1; i++)
        {
            for (int j = 0; full_name[i][j] != '\0'; j++)
            {
                if (!isalpha(full_name[i][j]) &&
full_name[i][j] != '_') {
                    return false;
                }-> if the characters of the string are not
letters between A and z and if the count of the underscore is other
than 1 it return false, so that user has to enter the name again.

    else if (count_underscore != 1) {
        return false;
    }
    return true; -> if the name check survived all of the steps above
it returns true, after which the user is required to enter the phone
number.


bool error_phone_number(char phone_number[L][N]) :

for (int i = 0; i < 1; i++)
    {
        for (int j = 0; phone_number[i][j] != '\0'; j++)
        {
            ++count_digit; -> counts the characters of the
entered string (including the plus sign)
        }
    }

if (phone_number[0][0] == '+') {
        for (int i = 0; i < 1; i++)
        {
            for (int j = 1; j < strlen(phone_number[i]); j++) ->
j=0 is ‘+’, that’s why j starts from 1
            {
                if (!(phone_number[i][j] >= '0' &&
phone_number[i][j] <= '9' && count_digit > 6)) {
                    return false; -> if the character
starting from the 2nd one is other than digits and the count of them
are not bigger than 6 (minimal number of digits in the phone number
according to StackOverflow) it returns false (the 1st character of
the string is ‘+’)
                }

    else if (phone_number[0][0] >= '0' && phone_number[0][0] <=
'9') {
        for (int i = 0; i < 1; i++)
        {
            for (int j = 1; j < strlen(phone_number[i]); j++)
            {
                if (!(phone_number[i][j] >= '0' &&
phone_number[i][j] <= '9' && count_digit > 6)) {
                    return false; -> if the string starts
with the number, not the plus sign again the same rule as in first
if repeats
                }

    else if (!(phone_number[0][0] >= '0' && phone_number[0][0] <=
'9')) {
        return false; -> if the first characters of the string is
not a number it also returns false
    }

    return true; -> if survived return true
```

Note: I could have just checked if the char is number or not while counting it.

```
bool error_e_mail(char e_mail[L][N]):

for (int i = 0; i < 1; i++)
    {
        int count_prefix = 0;

        for (int j = 0; e_mail[i][j] != '@'; j++)
        {
            if (isalnum(e_mail[0][0])) { -> if the first char is
either a number or a letter till the at sign, increment the amount
of chars in the prefix. If the prefix starts with a char other than
number or letter count_prefix will not increment and will return
false (last condition in the following for loop)
                ++count_prefix;
            }
        }


for (int i = 0; i < 1; i++)
    {
        int count_prefix = 0;

        for (int j = 0; e_mail[i][j] != '@'; j++)
        {
            ++count_prefix; -> counts number of chars of the
part until ‘@’ for which the minimal count is 2
        }

        for (int j = 0; e_mail[i][j] != '\0'; j++)
        {
            if (count_prefix >= 2) {
```

```
if (strstr(e_mail[i], "@gmail.com")) { -> if the string has this
domain
                           if (!((isalpha(e_mail[i][j])) ||
e_mail[i][j] == '_' || e_mail[i][j] == '.' || e_mail[i][j] == '@' ||
(e_mail[i][j] >= '0' && e_mail[i][j] <= '9'))) {
                                 return false; -> anything besides
letters, numbers, underscore, dot and at sign will return false
                           }
                           else if (e_mail[i][j] == '@') {
                                 if (e_mail[i][j + 10] != '\0') {
                                      return false; -> ‘\0’ has to
come 10 chars after the ‘@’, it checks whether the string ends with
gmail.com and prevents from having mistakes such as gmail.comm
                                 }


                      else if (strstr(e_mail[i], "@mail.ru")) {
                           if (!((isalpha(e_mail[i][j])) ||
e_mail[i][j] == '_' || e_mail[i][j] == '.' || e_mail[i][j] == '@' ||
(e_mail[i][j] >= '0' && e_mail[i][j] <= '9'))) {
                                 return false;
                           }
                           else if (e_mail[i][j] == '@') {
                                 if (e_mail[i][j + 8] != '\0') { ->
for mail.ru ‘\0’ comes 8 chars after ‘@’
                                      return false;
                                 }



                      else if (!(strstr(e_mail[i], "@gmail.com") ||
strstr(e_mail[i], "@mail.ru"))) {
                           return false; -> if non of these 2
domains are present in the string then it returns false;
                      }
                }
                else if (count_prefix < 2) {
                      return false; -> and if the prefix contains
less than 2 chars it returns false
                }

     return true; -> if survived true
```

Note: only gmail.com and mail.ru were used in this project, assuming that this was a phonebook of a person who lives in Azerbaijan (Fatima) and the most used ones are these and no one puts someone’s school/work e-mail into their contact information.

```
bool error_address(char contact_address[L][N]) :
```

```
for (int i = 0; i < 1; i++)
    {
        for (int j = 0; j < strlen(contact_address[i]); j++)
        {
            if (isalpha(contact_address[0][0]) ||
(contact_address[0][0] >= '0' && contact_address[0][0] <= '9'){ ->
first char has to be either letter or a number
                if (contact_address[i][j] == '_' && j !=
strlen(contact_address[i]) - 1) {
                    ++count_underscore; -> same underscore
count story as in full name, but here it also allows numbers to be
included


    if (count_underscore >= 1) { -> must have at least one
underscore in the middle
        for (int i = 0; i < 1; i++)
        {
            for (j = 0; contact_address[i][j] != '_'; j++)
            {
                if (!isdigit(contact_address[i][j]) &&
!isalpha(contact_address[i][j])) { -> string must contain only
numbers or letters until the first underscore
                    return false;
                }
            }
            for (int z = j; contact_address[i][z] != '\0'; z++)
            {
                if (!isalpha(contact_address[i][z]) &&
contact_address[i][z] != '_') { -> starting from the first char
after underscore string must contain only letters and underscores,
numbers must appear only at the very beginning of the string (until
the first underscore)
                    return false;
                }

    else if (count_underscore < 1) {
        return false; -> if there is no underscore returns false
}
    }bool error_work(char work[L][N]), bool error_relationship(char
relationship[L][N])  :

for (int i = 0; i < 1; i++)
    {
        for (int j = 0; j < strlen(work[i]); j++)
        {
            if (!isalpha(work[i][j])){
                return false; -> work and relationship can
only contain letters
```

```
            }
        }

        return true; -> if survived return true
    }
```

void list_all() :

```
for (int i = 0; i < count; i++) -> till count to list all contacts’
information by attributes including the added ones
    {
        printf("\t\t\t\t\t\t\t\t\t\t\t  Contact's name: %s\n",
full_name[i]);
        printf("\t\t\t\t\t\t\t\t\t\t\t  Contact's phone number:
%s\n", phone_number[i]);
        printf("\t\t\t\t\t\t\t\t\t\t\t  Contact's e-mail: %s\n",
e_mail[i]);
        printf("\t\t\t\t\t\t\t\t\t\t\t  Contact's address: %s\n",
contact_address[i]);
        printf("\t\t\t\t\t\t\t\t\t\t\t  Contact's job: %s\n",
work[i]);
        printf("\t\t\t\t\t\t\t\t\t\t\t  Contact's relationship to
you: %s\n\n", relationship[i]);
    }
```

void total_contact() :

```
printf("\t\t\t\t\t\t\t\t\t\t\t  Total amount of contacts: %d\n",
count); -> to get the total amount of contacts we just need to print
the value of count (because we already incremented it in add
function, so it has the right info about amount of contacts in the
phonebook)
```

void search() :

```
char contact_for_search[50];
    int i;
    printf("\t\t\t\t\t\t\t\t\t\t\t  Enter the full name, phone number
or e-mail of the contact: ");
    scanf("%s", &contact_for_search);

    for (i = 0; i < L; i++) -> has to search all the strings
including test cases until L
    {
        if (strcmp(contact_for_search, full_name[i]) == 0) { ->
with strcmp we see if the entered string matches with any of the
existing
            printf("\t\t\t\t\t\t\t\t\t\t\t  Contact's phone
number: %s\n", phone_number[i]);
```

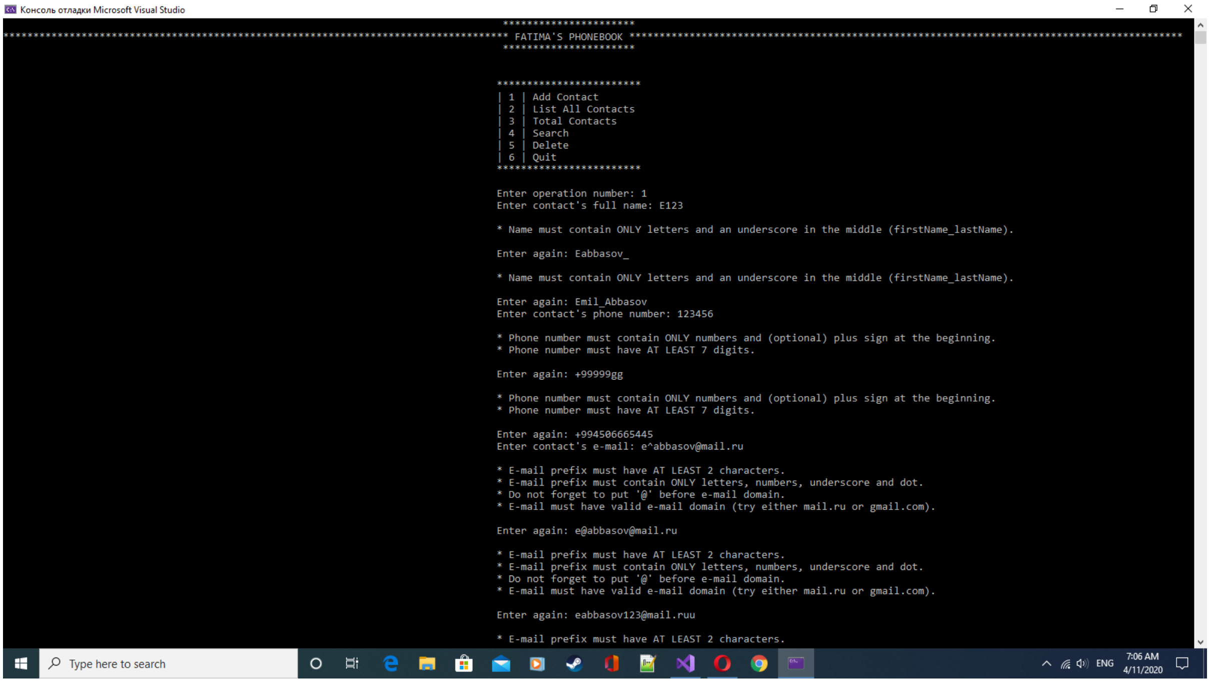
```
                printf("\t\t\t\t\t\t\t\t\t\t  Contact's e-mail: 
%s\n", e_mail[i]);
                printf("\t\t\t\t\t\t\t\t\t\t  Contact's address: 
%s\n", contact_address[i]);
                printf("\t\t\t\t\t\t\t\t\t\t  Contact's job: %s\n", 
work[i]);
                printf("\t\t\t\t\t\t\t\t\t\t  Contact's relationship 
to you: %s\n\n", relationship[i]);
                break; -> break once the contact is found
           }

else if (strcmp(contact_for_search, full_name[i]) != 0) {
                for (j = 0; contact_for_search[j] != '\0'; j++)
                {
                      if (contact_for_search[j] - full_name[i][j] == 
32 || full_name[i][j] - contact_for_search[j] == 32) {
                           continue; -> if the user puts f instead 
of F it still searches for the contact: difference is 32 because 
between every capital and non-capital letter there are 32 chars in 
between in ASCII table.
                      }
                      else if (contact_for_search[j] - 
full_name[i][j] == 0) { -> some characters still can be the same 
(instead of Fatima_Khalifali user enters Fatima_KhAlifali). It 
continues when it sees the same letters.
                           continue;
                      }
                      else {
                           break; -> if there is any other option, 
then it does not need to be checked, break the loop, the contact you 
are searching is not the one whose characters are being iterated.
                      }
                }

                if (full_name[i][j] == '\0' && contact_for_search[j] 
== '\0') { -> if program successfully finds the name matching the 
users search request then their j s stopped when they reached ‘\0’
                      printf("\t\t\t\t\t\t\t\t\t\t  Contact's phone 
number: %s\n", phone_number[i]);
                      printf("\t\t\t\t\t\t\t\t\t\t  Contact's e-
mail: %s\n", e_mail[i]);
                      printf("\t\t\t\t\t\t\t\t\t\t  Contact's 
address: %s\n", contact_address[i]);
                      printf("\t\t\t\t\t\t\t\t\t\t  Contact's job: 
%s\n", work[i]);
                      printf("\t\t\t\t\t\t\t\t\t\t  Contact's 
relationship to you: %s\n\n", relationship[i]);
                      break;
                }
```

```
            else if (i == L) { -> if the program can not find
the requested name i will iterate throughout all the test cases and
added contacts in full_name[L][N] so if it finally is L without
breaking then there is no such contact
                printf("\t\t\t\t\t\t\t\t\t\t  Contact NOT
FOUND.\n\n");
                break;
            }
    }

    if (i == L) {
        printf("\t\t\t\t\t\t\t\t\t\t  Contact NOT FOUND.\n\n");->
if i end up being equal to L it means the loop iterated till the end
and did not find (break) the appropriate contact
    }
```

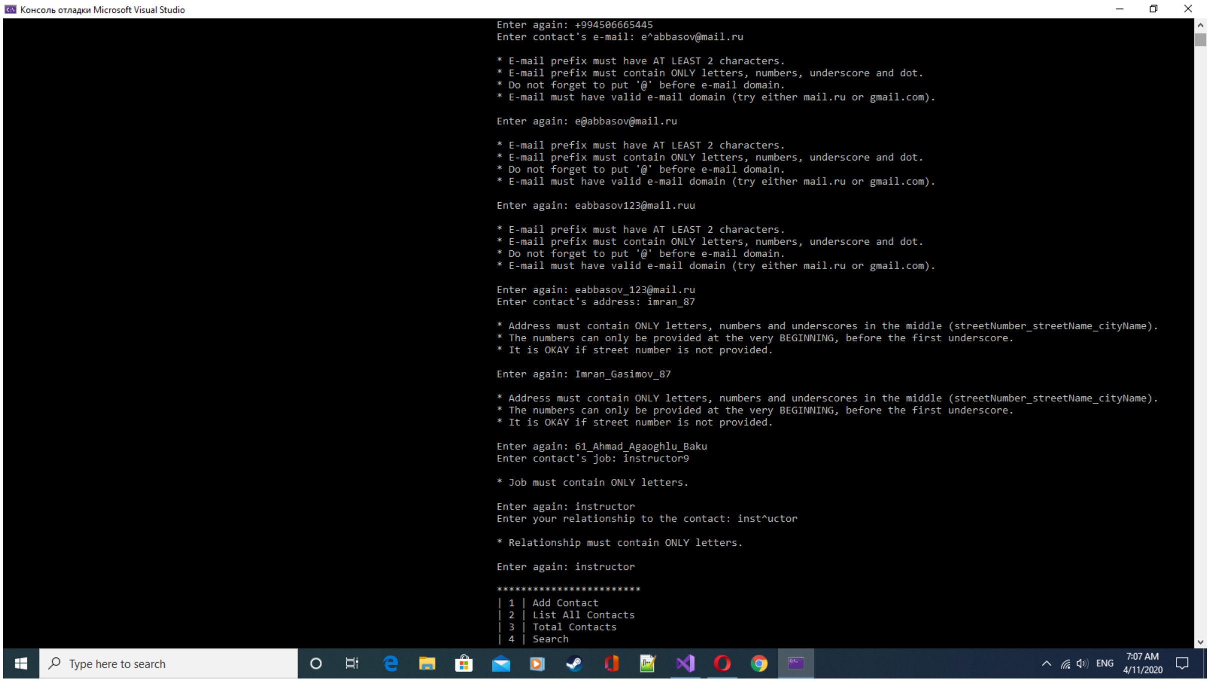
Note: 3 unique attributes are presented; 2 of them (phone number and e-mail) are undoubtedly unique for every contact, however full name can be same sometimes, but as long as in phonebooks we do not assign same name to both people who have same full names, full name is treated as a unique attribute. And also the attribute the user searches by does not appear in the information provided afterwards. Phone number and e-mail search are not copied to this report, because they work as the full name only without else if which allows user to make capital and non-capital letter mistakes. (Phone number consists of only numbers and e-mail does not tolerate such letter mistakes, it has to be precise.)

void delete() :

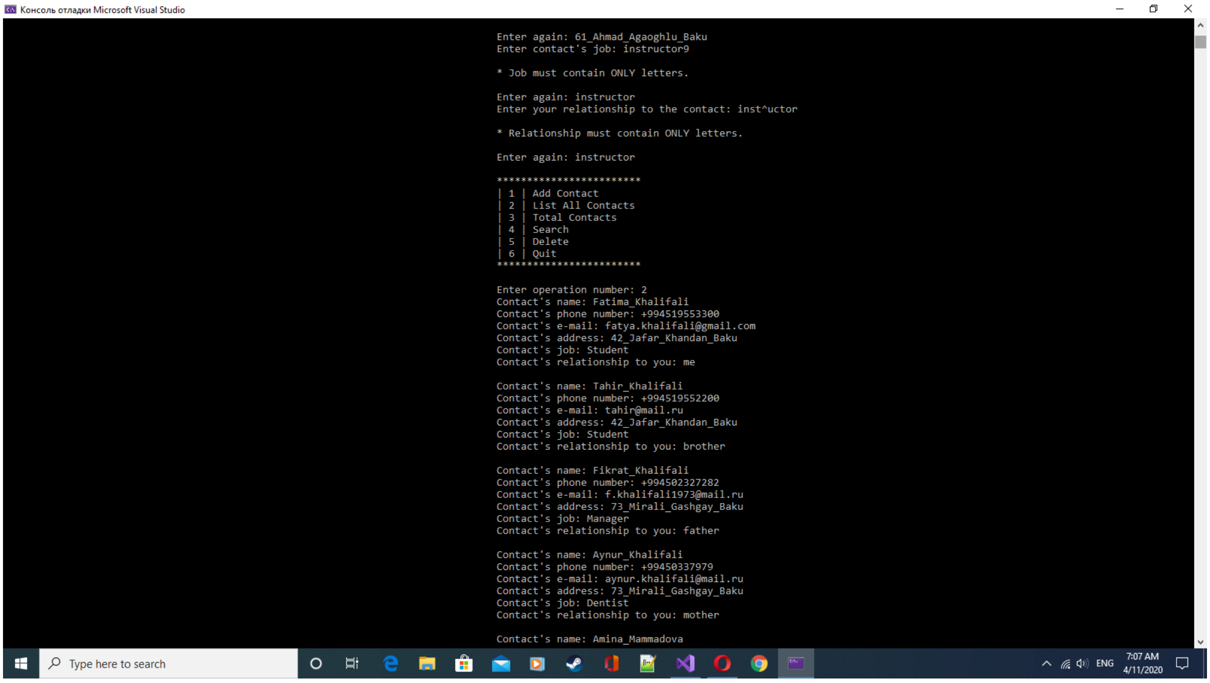
```
    char contact_for_search[50];
    int i;
    printf("\t\t\t\t\t\t\t\t\t\t  Enter the full name, phone number
or e-mail of the contact: ");
    scanf("%s", &contact_for_search);

    for (i = 0; i < L; i++)
    {
        if (strcmp(contact_for_search, full_name[i]) == 0 ||
strcmp(contact_for_search, phone_number[i]) == 0 ||
strcmp(contact_for_search, e_mail[i]) == 0) { -> if the user wrote
everything precisely the program will start searching here and break
when it finds appropriate contact
            break;
        }else if (strcmp(contact_for_search, full_name[i]) != 0)
-> if the user did not enter full name of the contact precisely  {
            for (j = 0; contact_for_search[j] != '\0'; j++)
            {
                if (contact_for_search[j] - full_name[i][j] ==
32 || full_name[i][j] - contact_for_search[j] == 32) {
                    continue;
                }
                else if (contact_for_search[j] -
full_name[i][j] == 0) {
                    continue;
                }
                else {
                    break;
                }
            }
            if (contact_for_search[j] == '\0' && full_name[i][j]
== '\0') {
                break; -> if it has found the right string it
has to break

if (i == L) {
        printf("\t\t\t\t\t\t\t\t\t\t  Contact NOT FOUND.\n\n");

else if (i != L) {
        for (int place = i; place < L - 1; place++) -> place is
equal to i meaning that i is the string which the user searches for.
And it checks till L-1 because L does not need to be checked (when
you are in L-1 you are going to replace L – 1 with L).
        {
            for (int j = 0; j < N; j++) -> N is 100 so it will
check the largest strings without causing problems
            {
                full_name[place][j] = full_name[place + 1][j];
                phone_number[place][j] = phone_number[place +
1][j];

                e_mail[place][j] = e_mail[place + 1][j];
                contact_address[place][j] =
contact_address[place + 1][j];
                work[place][j] = work[place + 1][j];
                relationship[place][j] = relationship[place +
1][j];
            } -> every attribute changes it place by going 1
slot to the left
        }
        count--; -> as the user deletes, count has to decrease.
It has to decrease inside else if function, otherwise even if the
contact is not found the total amount will decrease and show the
wrong number of contacts.
```



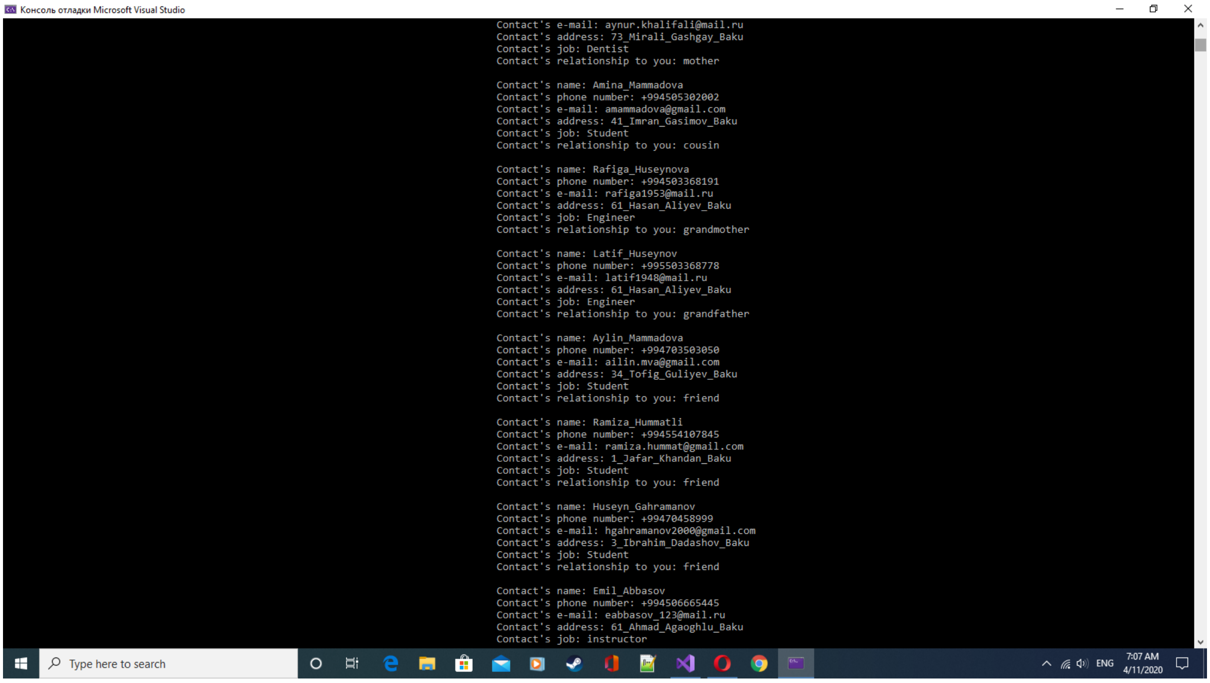
```
                                                                                    *********************
************************************************************************************ FATIMA'S PHONEBOOK ***************************************************************************************************
                                                                                    *********************


                                                                                   ************************
                                                                                   | 1 | Add Contact
                                                                                   | 2 | List All Contacts
                                                                                   | 3 | Total Contacts
                                                                                   | 4 | Search
                                                                                   | 5 | Delete
                                                                                   | 6 | Quit
                                                                                   ************************

                                                                                   Enter operation number: 1
                                                                                   Enter contact's full name: E123

                                                                                   * Name must contain ONLY letters and an underscore in the middle (firstName_lastName).

                                                                                   Enter again: Eabbasov_

                                                                                   * Name must contain ONLY letters and an underscore in the middle (firstName_lastName).

                                                                                   Enter again: Emil_Abbasov
                                                                                   Enter contact's phone number: 123456

                                                                                   * Phone number must contain ONLY numbers and (optional) plus sign at the beginning.
                                                                                   * Phone number must have AT LEAST 7 digits.

                                                                                   Enter again: +99999gg

                                                                                   * Phone number must contain ONLY numbers and (optional) plus sign at the beginning.
                                                                                   * Phone number must have AT LEAST 7 digits.

                                                                                   Enter again: +994506665445
                                                                                   Enter contact's e-mail: e^abbasov@mail.ru

                                                                                   * E-mail prefix must have AT LEAST 2 characters.
                                                                                   * E-mail prefix must contain ONLY letters, numbers, underscore and dot.
                                                                                   * Do not forget to put '@' before e-mail domain.
                                                                                   * E-mail must have valid e-mail domain (try either mail.ru or gmail.com).

                                                                                   Enter again: e@abbasov@mail.ru

                                                                                   * E-mail prefix must have AT LEAST 2 characters.
                                                                                   * E-mail prefix must contain ONLY letters, numbers, underscore and dot.
                                                                                   * Do not forget to put '@' before e-mail domain.
                                                                                   * E-mail must have valid e-mail domain (try either mail.ru or gmail.com).

                                                                                   Enter again: eabbasov123@mail.ruu

                                                                                   * E-mail prefix must have AT LEAST 2 characters.
```



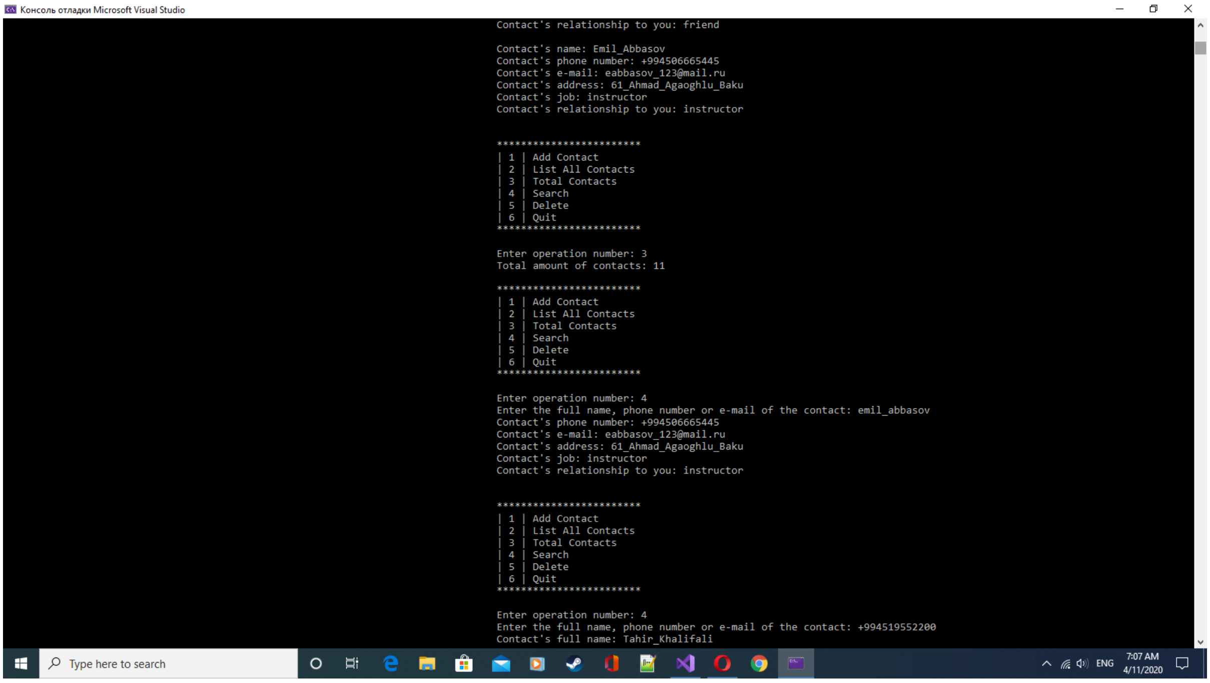
```
Enter again: +994506665445
Enter contact's e-mail: e^abbasov@mail.ru

* E-mail prefix must have AT LEAST 2 characters.
* E-mail prefix must contain ONLY letters, numbers, underscore and dot.
* Do not forget to put '@' before e-mail domain.
* E-mail must have valid e-mail domain (try either mail.ru or gmail.com).

Enter again: e@abbasov@mail.ru

* E-mail prefix must have AT LEAST 2 characters.
* E-mail prefix must contain ONLY letters, numbers, underscore and dot.
* Do not forget to put '@' before e-mail domain.
* E-mail must have valid e-mail domain (try either mail.ru or gmail.com).

Enter again: eabbasov123@mail.ruu

* E-mail prefix must have AT LEAST 2 characters.
* E-mail prefix must contain ONLY letters, numbers, underscore and dot.
* Do not forget to put '@' before e-mail domain.
* E-mail must have valid e-mail domain (try either mail.ru or gmail.com).

Enter again: eabbasov_123@mail.ru
Enter contact's address: imran_87

* Address must contain ONLY letters, numbers and underscores in the middle (streetNumber_streetName_cityName).
* The numbers can only be provided at the very BEGINNING, before the first underscore.
* It is OKAY if street number is not provided.

Enter again: Imran_Gasimov_87

* Address must contain ONLY letters, numbers and underscores in the middle (streetNumber_streetName_cityName).
* The numbers can only be provided at the very BEGINNING, before the first underscore.
* It is OKAY if street number is not provided.

Enter again: 61_Ahmad_Agaoghlu_Baku
Enter contact's job: instructor9

* Job must contain ONLY letters.

Enter again: instructor
Enter your relationship to the contact: inst^uctor

* Relationship must contain ONLY letters.

Enter again: instructor

************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
```



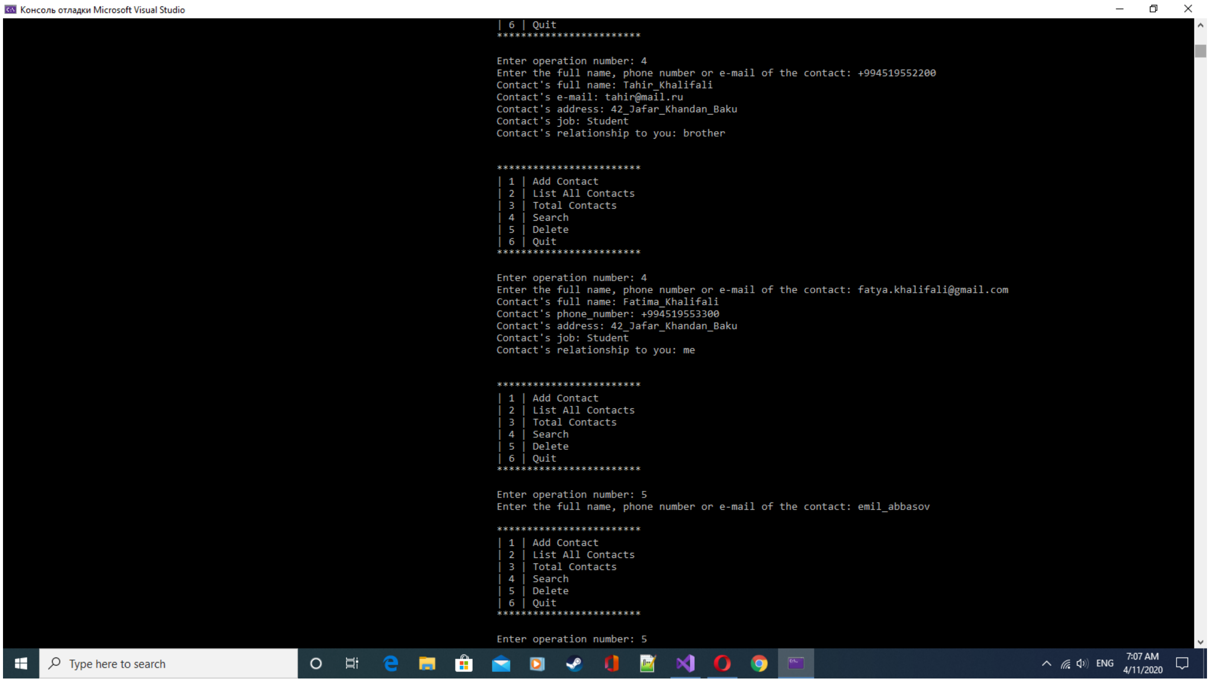
```
Enter again: 61_Ahmad_Agaoghlu_Baku
Enter contact's job: instructor9

* Job must contain ONLY letters.

Enter again: instructor
Enter your relationship to the contact: inst^uctor

* Relationship must contain ONLY letters.

Enter again: instructor

************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 2
Contact's name: Fatima_Khalifali
Contact's phone number: +994519553300
Contact's e-mail: fatya.khalifali@gmail.com
Contact's address: 42_Jafar_Khandan_Baku
Contact's job: Student
Contact's relationship to you: me

Contact's name: Tahir_Khalifali
Contact's phone number: +994519552200
Contact's e-mail: tahir@mail.ru
Contact's address: 42_Jafar_Khandan_Baku
Contact's job: Student
Contact's relationship to you: brother

Contact's name: Fikrat_Khalifali
Contact's phone number: +994502327282
Contact's e-mail: f.khalifali1973@mail.ru
Contact's address: 73_Mirali_Gashgay_Baku
Contact's job: Manager
Contact's relationship to you: father

Contact's name: Aynur_Khalifali
Contact's phone number: +99450337979
Contact's e-mail: aynur.khalifali@mail.ru
Contact's address: 73_Mirali_Gashgay_Baku
Contact's job: Dentist
Contact's relationship to you: mother

Contact's name: Amina_Mammadova
```



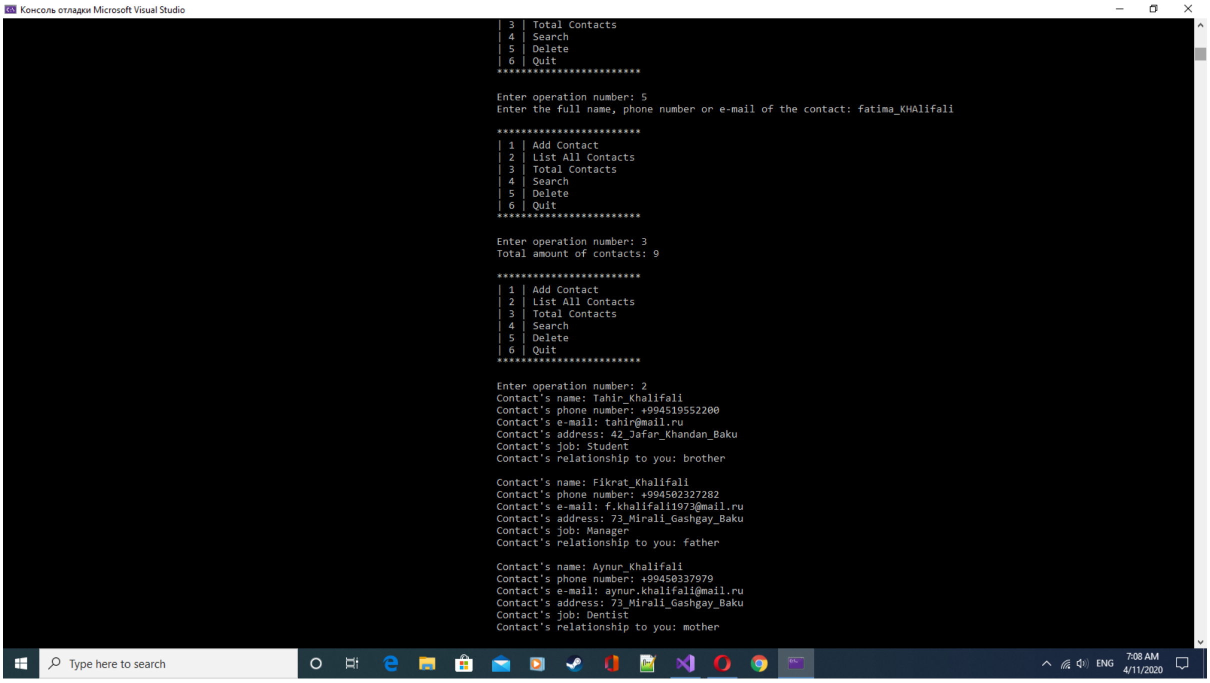
```
Contact's e-mail: aynur.khalifali@mail.ru
Contact's address: 73_Mirali_Gashgay_Baku
Contact's job: Dentist
Contact's relationship to you: mother

Contact's name: Amina_Mammadova
Contact's phone number: +994505302002
Contact's e-mail: amammadova@gmail.com
Contact's address: 41_Imran_Gasimov_Baku
Contact's job: Student
Contact's relationship to you: cousin

Contact's name: Rafiga_Huseynova
Contact's phone number: +994503368191
Contact's e-mail: rafiga1953@mail.ru
Contact's address: 61_Hasan_Aliyev_Baku
Contact's job: Engineer
Contact's relationship to you: grandmother

Contact's name: Latif_Huseynov
Contact's phone number: +995503368778
Contact's e-mail: latif1948@mail.ru
Contact's address: 61_Hasan_Aliyev_Baku
Contact's job: Engineer
Contact's relationship to you: grandfather

Contact's name: Aylin_Mammadova
Contact's phone number: +994703503050
Contact's e-mail: ailin.mva@gmail.com
Contact's address: 34_Tofig_Guliyev_Baku
Contact's job: Student
Contact's relationship to you: friend

Contact's name: Ramiza_Hummatli
Contact's phone number: +994554107845
Contact's e-mail: ramiza.hummat@gmail.com
Contact's address: 1_Jafar_Khandan_Baku
Contact's job: Student
Contact's relationship to you: friend

Contact's name: Huseyn_Gahramanov
Contact's phone number: +99470458999
Contact's e-mail: hgahramanov2000@gmail.com
Contact's address: 3_Ibrahim_Dadashov_Baku
Contact's job: Student
Contact's relationship to you: friend

Contact's name: Emil_Abbasov
Contact's phone number: +994506665445
Contact's e-mail: eabbasov_123@mail.ru
Contact's address: 61_Ahmad_Agaoghlu_Baku
Contact's job: instructor
```



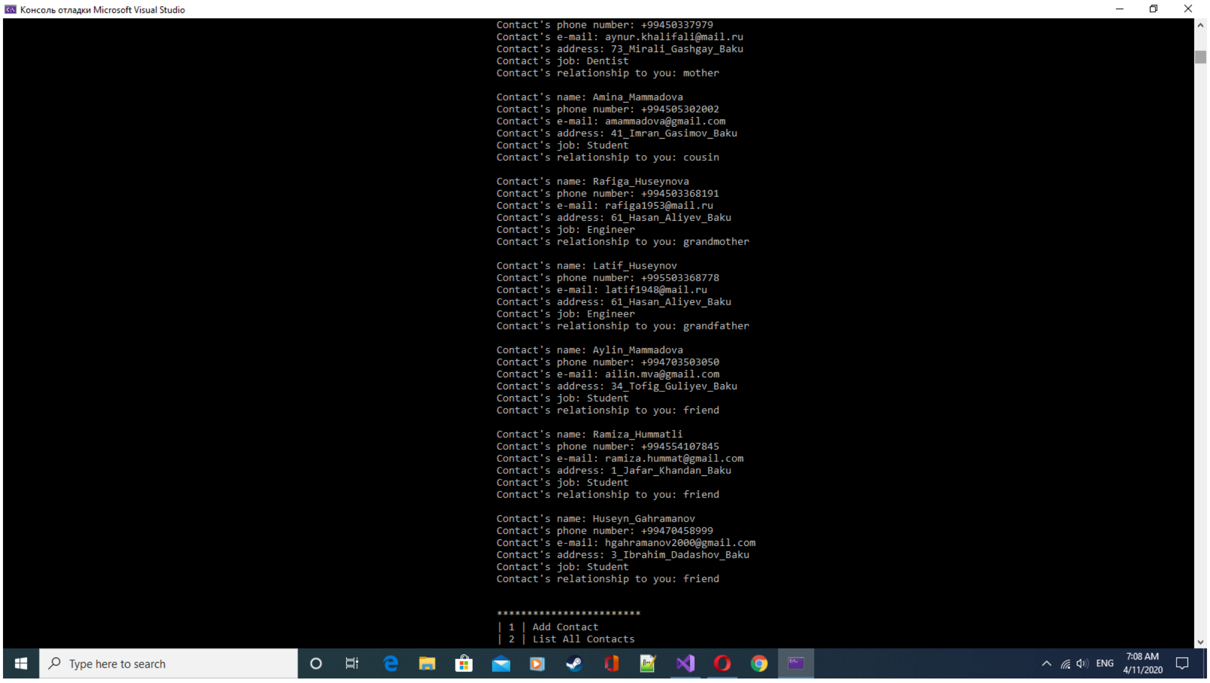
```
Contact's relationship to you: friend

Contact's name: Emil_Abbasov
Contact's phone number: +994506665445
Contact's e-mail: eabbasov_123@mail.ru
Contact's address: 61_Ahmad_Agaoghlu_Baku
Contact's job: instructor
Contact's relationship to you: instructor


************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 3
Total amount of contacts: 11

************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 4
Enter the full name, phone number or e-mail of the contact: emil_abbasov
Contact's phone number: +994506665445
Contact's e-mail: eabbasov_123@mail.ru
Contact's address: 61_Ahmad_Agaoghlu_Baku
Contact's job: instructor
Contact's relationship to you: instructor


************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 4
Enter the full name, phone number or e-mail of the contact: +994519552200
Contact's full name: Tahir_Khalifali
```



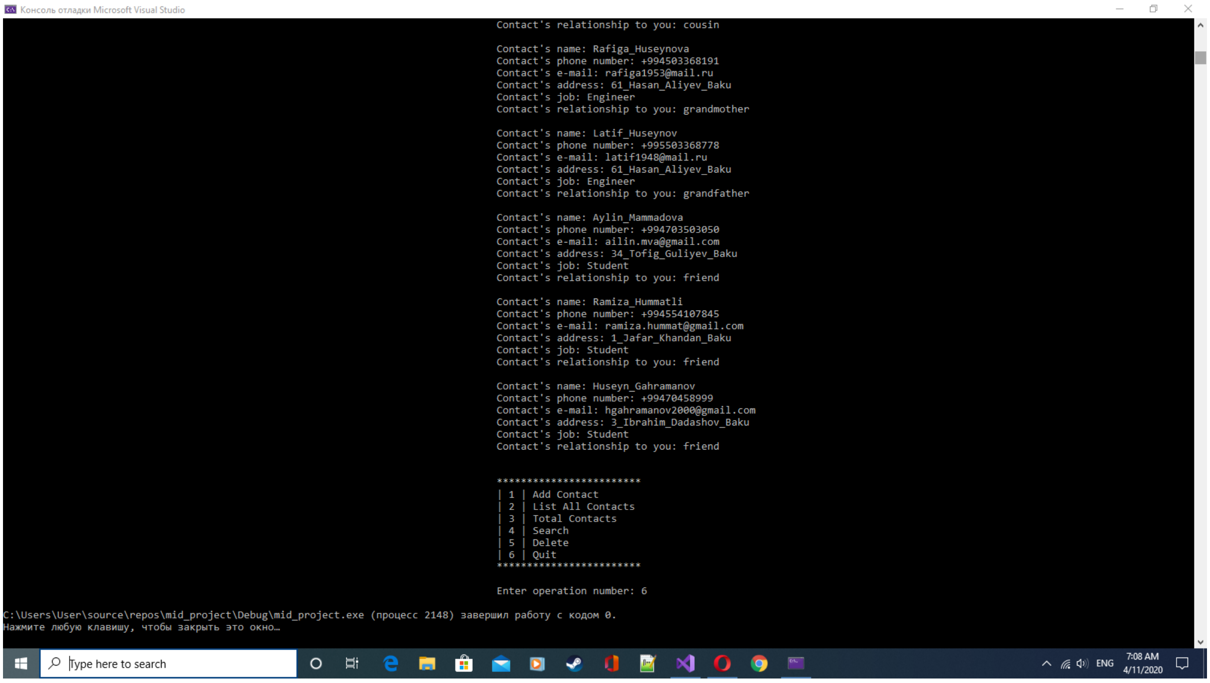
```
| 6 | Quit
************************

Enter operation number: 4
Enter the full name, phone number or e-mail of the contact: +994519552200
Contact's full name: Tahir_Khalifali
Contact's e-mail: tahir@mail.ru
Contact's address: 42_Jafar_Khandan_Baku
Contact's job: Student
Contact's relationship to you: brother


************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 4
Enter the full name, phone number or e-mail of the contact: fatya.khalifali@gmail.com
Contact's full name: Fatima_Khalifali
Contact's phone_number: +994519553300
Contact's address: 42_Jafar_Khandan_Baku
Contact's job: Student
Contact's relationship to you: me


************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 5
Enter the full name, phone number or e-mail of the contact: emil_abbasov

************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 5
```



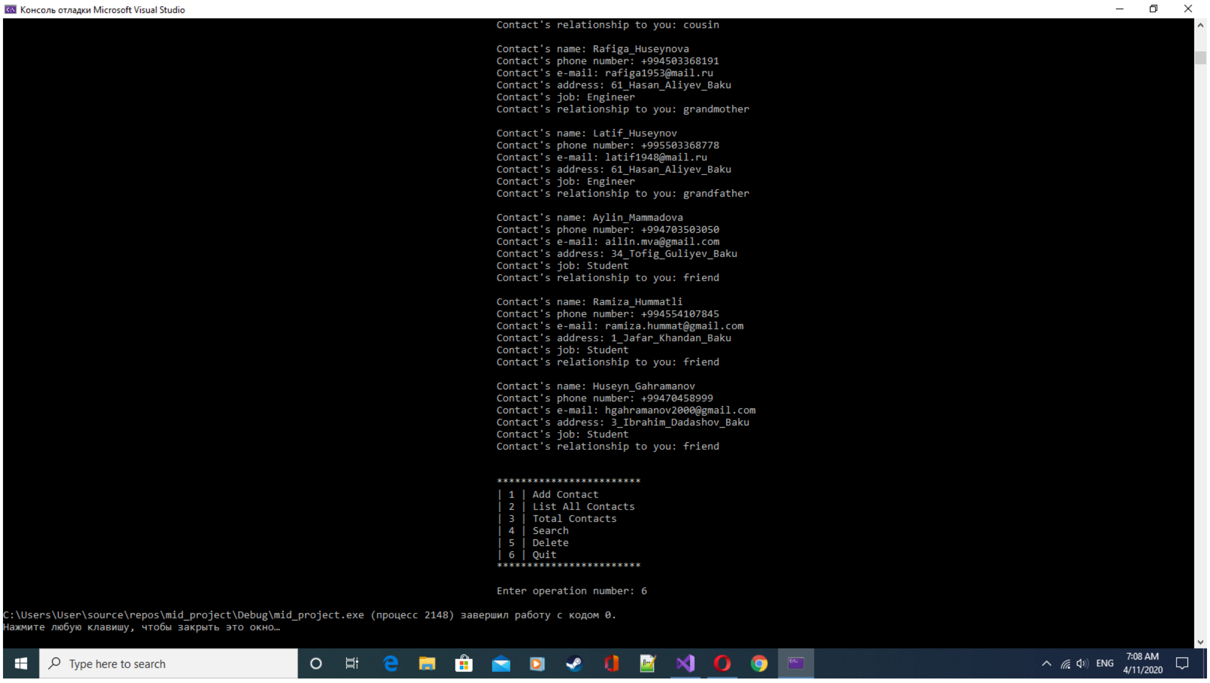
```
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 5
Enter the full name, phone number or e-mail of the contact: fatima_KHAlifali

************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 3
Total amount of contacts: 9

************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 2
Contact's name: Tahir_Khalifali
Contact's phone number: +994519552200
Contact's e-mail: tahir@mail.ru
Contact's address: 42_Jafar_Khandan_Baku
Contact's job: Student
Contact's relationship to you: brother

Contact's name: Fikrat_Khalifali
Contact's phone number: +994502327282
Contact's e-mail: f.khalifali1973@mail.ru
Contact's address: 73_Mirali_Gashgay_Baku
Contact's job: Manager
Contact's relationship to you: father

Contact's name: Aynur_Khalifali
Contact's phone number: +99450337979
Contact's e-mail: aynur.khalifali@mail.ru
Contact's address: 73_Mirali_Gashgay_Baku
Contact's job: Dentist
Contact's relationship to you: mother
```



```
Contact's phone number: +99450337979
Contact's e-mail: aynur.khalifali@mail.ru
Contact's address: 73_Mirali_Gashgay_Baku
Contact's job: Dentist
Contact's relationship to you: mother

Contact's name: Amina_Mammadova
Contact's phone number: +994505302002
Contact's e-mail: amammadova@gmail.com
Contact's address: 41_Imran_Gasimov_Baku
Contact's job: Student
Contact's relationship to you: cousin

Contact's name: Rafiga_Huseynova
Contact's phone number: +994503368191
Contact's e-mail: rafiga1953@mail.ru
Contact's address: 61_Hasan_Aliyev_Baku
Contact's job: Engineer
Contact's relationship to you: grandmother

Contact's name: Latif_Huseynov
Contact's phone number: +995503368778
Contact's e-mail: latif1948@mail.ru
Contact's address: 61_Hasan_Aliyev_Baku
Contact's job: Engineer
Contact's relationship to you: grandfather

Contact's name: Aylin_Mammadova
Contact's phone number: +994703503050
Contact's e-mail: ailin.mva@gmail.com
Contact's address: 34_Tofig_Guliyev_Baku
Contact's job: Student
Contact's relationship to you: friend

Contact's name: Ramiza_Hummatli
Contact's phone number: +994554107845
Contact's e-mail: ramiza.hummat@gmail.com
Contact's address: 1_Jafar_Khandan_Baku
Contact's job: Student
Contact's relationship to you: friend

Contact's name: Huseyn_Gahramanov
Contact's phone number: +99470458999
Contact's e-mail: hgahramanov2000@gmail.com
Contact's address: 3_Ibrahim_Dadashov_Baku
Contact's job: Student
Contact's relationship to you: friend


************************
| 1 | Add Contact
| 2 | List All Contacts
```



```
Contact's relationship to you: cousin

Contact's name: Rafiga_Huseynova
Contact's phone number: +994503368191
Contact's e-mail: rafiga1953@mail.ru
Contact's address: 61_Hasan_Aliyev_Baku
Contact's job: Engineer
Contact's relationship to you: grandmother

Contact's name: Latif_Huseynov
Contact's phone number: +995503368778
Contact's e-mail: latif1948@mail.ru
Contact's address: 61_Hasan_Aliyev_Baku
Contact's job: Engineer
Contact's relationship to you: grandfather

Contact's name: Aylin_Mammadova
Contact's phone number: +994703503050
Contact's e-mail: ailin.mva@gmail.com
Contact's address: 34_Tofig_Guliyev_Baku
Contact's job: Student
Contact's relationship to you: friend

Contact's name: Ramiza_Hummatli
Contact's phone number: +994554107845
Contact's e-mail: ramiza.hummat@gmail.com
Contact's address: 1_Jafar_Khandan_Baku
Contact's job: Student
Contact's relationship to you: friend

Contact's name: Huseyn_Gahramanov
Contact's phone number: +99470458999
Contact's e-mail: hgahramanov2000@gmail.com
Contact's address: 3_Ibrahim_Dadashov_Baku
Contact's job: Student
Contact's relationship to you: friend


************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 6

C:\Users\User\source\repos\mid_project\Debug\mid_project.exe (процесс 2148) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```



Консоль отладки Microsoft Visual Studio

```
Contact's relationship to you: cousin

Contact's name: Rafiga_Huseynova
Contact's phone number: +994503368191
Contact's e-mail: rafiga1953@mail.ru
Contact's address: 61_Hasan_Aliyev_Baku
Contact's job: Engineer
Contact's relationship to you: grandmother

Contact's name: Latif_Huseynov
Contact's phone number: +995503368778
Contact's e-mail: latif1948@mail.ru
Contact's address: 61_Hasan_Aliyev_Baku
Contact's job: Engineer
Contact's relationship to you: grandfather

Contact's name: Aylin_Mammadova
Contact's phone number: +994703503050
Contact's e-mail: ailin.mva@gmail.com
Contact's address: 34_Tofig_Guliyev_Baku
Contact's job: Student
Contact's relationship to you: friend

Contact's name: Ramiza_Hummatli
Contact's phone number: +994554107845
Contact's e-mail: ramiza.hummat@gmail.com
Contact's address: 1_Jafar_Khandan_Baku
Contact's job: Student
Contact's relationship to you: friend

Contact's name: Huseyn_Gahramanov
Contact's phone number: +99470458999
Contact's e-mail: hgahramanov2000@gmail.com
Contact's address: 3_Ibrahim_Dadashov_Baku
Contact's job: Student
Contact's relationship to you: friend


************************
| 1 | Add Contact
| 2 | List All Contacts
| 3 | Total Contacts
| 4 | Search
| 5 | Delete
| 6 | Quit
************************

Enter operation number: 6

C:\Users\User\source\repos\mid_project\Debug\mid_project.exe (процесс 2148) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

Type here to search
ENG
7:08 AM
4/11/2020