```
/*
Copyright 2015 ALY SHMAHELL

 This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Lesser General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.
*/
#include <iostream>
#include <string.h>
#include <sstream>
#include <stdio.h>
#include <stdint.h>
#include "ENDABI_RSA_CORE.hpp"
#include "3rd_party/InfInt.h"

using namespace std;
int main()
{
    InfInt desired_pk,desired_p,desired_q;
    cout<<"please input the desired public-key sequence \n(desired public-key, desir
ed p, desired q) and wait for us to make the right \ncorrections and generate a suit
able public-key\n>> ";
    cin>>desired_pk>>desired_p>>desired_q;
    pub_key_sequence<InfInt> pks = generate_public_key(desired_pk,desired_p,desired_
q);
    stringstream pks_string;
    pks_string<<"public-key sequence : pk: "<<pks.pk<<"  n:  "<<pks.n;
    cout<<pks_string.str()<<endl;
    cout<<"do you want a private key? Y or N\n";
    string answer;
    cin>>answer;
    if(answer=="Y"||answer=="y")
    {
        InfInt prvk = calculate_private_key(pks.pk,pks.p_,pks.q_);
        stringstream prvks_string;
        prvks_string<<"private-key : "<<prvk<<" n: "<<pks.n;
        cout<<prvks_string.str()<<endl;
        printf("Do you want to encrypt a numerical message? \nType either Y or N.\n>
> ");
        cin>>answer;
        if(answer=="Y"||answer=="y")
        {
        printf("Good, now we can continue,\nIf typed Y please input the desired mess
age (max char count is 200) :\n>> ");
            char message [200];
            InfInt plain_numerical[200];
            InfInt encrypted_numerical[200];
            InfInt decrypted_numerical[200];
            cin.ignore();
            gets(message);
            int mssg_length = strlen(message);
            for(int i =0; i<mssg_length; i++)
            {
                plain_numerical[i]=message[i]-'0';
                encrypted_numerical[i]=encrypt((InfInt)plain_numerical[i],pks.pk,pks
.n);
                cout<<encrypted_numerical[i];
            }
            printf("\n");
            cout<<"Now, do you want to decrypt the message?\n>> ";
                cin>>answer;
            if(answer=="Y"||answer=="y")
            {
                for(int i =0; i<mssg_length; i++)
                {
                    decrypted_numerical[i]=decrypt(encrypted_numerical[i],prvk,pks.n
);
                    cout<<decrypted_numerical[i];
                }
                printf("\n");
            }
            else
            {
                printf("Well, there's nothing more we can do for you then.\nHappy in
securty :)\n");
                return 0;
            }
        }
        else
        {
            printf("Well, there's nothing more we can do for you then.\nHappy insecu
rty :)\n");
            return 0;
        }
    }
    else
    {
        printf("Well, there's nothing more we can do for you then.\nHappy insecurty
:)\n");
        return 0;
    }

}
```