

**1. Read the Data**

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <string>

using namespace std;

struct Information
{
    string firstname, lastname, password;
    int ssn, day, month, year;
};

string normalize(string name)
{
    string result = "";
    int i = 0, length = name.length(); while (i < length)
    {
        char c = name[i];
        if (islower(c))
            result = result + c;
        else
            result = result + (char)tolower(c);
        i = i + 1;
    } return result;
}

void readfile(Information info[])
{
    ifstream in;
    in.open("/home/118/database1.txt");
    if (in.fail())
    {
        cout << "Cannot read the file\n";
        exit(1);
    }
    int pos = 0;
    while (true)
    {
```

```

        in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
        >> info[pos].day >> info[pos].month >> info[pos].year
        >> info[pos].password; if (in.fail())
            break;
        pos = pos + 1;
    }
    in.close();
    pos = 0;
    while (pos < 10)
    {
        cout << info[pos].firstname << " " << info[pos].lastname
            << " " << info[pos].ssn << " " << info[pos].day
            << " " << info[pos].month << " " << info[pos].year
            << " " << info[pos].year << " " << info[pos].password << "\n";
        pos = pos + 1;
    }
    pos = 990;
    while (pos <= 999)
    {
        cout << info[pos].firstname << " " << info[pos].lastname
            << " " << info[pos].ssn << " " << info[pos].day
            << " " << info[pos].month << " " << info[pos].year
            << " " << info[pos].year << " " << info[pos].password << "\n";
        pos = pos + 1;
    }
}

int main()
{
    Information info[1000];
    readfile(info);
    string name;
}

```

The screenshot shows a Windows 11 desktop with a taskbar at the bottom. A Windows PowerShell window is open, displaying the following C++ code and its output:

```

<< " " << info[pos].year << " " << info[pos].password << "\n";
pos = pos + 1;
}

int main()
{
    Information info[1000];
    readfile(info);
    string name;
}

sma457@rabbit:~$ g++ 11.cpp
sma457@rabbit:~$ ./a.out

```

The output of the program is a list of 1000 records, each containing a name, a social security number (ssn), a day, a month, a year, and a password. The records are displayed in a single column, with each record on a new line. The records are as follows:

Betty Lamprey	181440880	10	8	1957	1957	XaLSKwLe
Pinky Fallowes	183880155	24	6	1927	1927	v6ysEyJV
Oliver Wriggle	186180093	14	3	1947	1947	ndmBmcV
Larry Brown	188690448	5	1	1948	1948	4n64bKde
Adolf Davies	111180924	28	10	1964	1964	10FtoODI
Bella Napster	112280747	27	2	1971	1971	H0Bemy8z
Marianne Stone	112580407	16	7	1973	1973	zcZzKile
Apu Mitchell	113078579	4	5	1957	1957	ce9bnaVv
Jilly Aston	113220519	23	10	1948	1948	0xH0SeyD
Matilda Vincent	114680858	7	6	1967	1967	2nFclvBa
Cora Wallace	992368644	4	6	1928	1928	8oIpALl9
Fido Billingsley	993100862	2	8	1961	1961	Dl8tK5Gm
Darrell Condon	993460662	25	3	1966	1966	axMezEH0
Norbert Cooke	994638286	7	11	1938	1938	CqEK4B8P
Sigmund Jansson	995878376	30	9	1982	1982	wPpLUXBX
Beatrice Abbot	998880533	7	2	1941	1941	5LfEEbeD
Curtis Croydon	999180994	11	1	1967	1967	wQZVlULN
Gary Reese	999180938	19	6	1978	1978	LeVUCPvH
Ebola Lester	999248067	15	2	1937	1937	B8XVVvH0
Devon Paulson	999430841	19	5	1967	1967	XHggkLkT8

## 2. Basic Search

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <string>

using namespace std;

struct Information
{
    string firstname, lastname, password;
    int ssn, day, month, year;
};

string normalize(string name)
{
    string result = "";
    int i = 0, length = name.length(); while (i < length)
    {
        char c = name[i];
        if (islower(c))
            result = result + c;
        else
            result = result + (char)tolower(c);
        i = i + 1;
    } return result;
}

void search(string name, Information info[]) {
    int pos = 0;
    int test = 0;
    while (pos <= 999)
    {
        string info1 = normalize(info[pos].firstname);
        string info2 = normalize(info[pos].lastname); string name1 = normalize(name);
        if (info1 == name1 || info2 == name1)
        {
            cout << info[pos].firstname << " " << info[pos].lastname << " "
                << info[pos].ssn << " " << info[pos].day << " " << info[pos].month
                << " " << info[pos].year << " " << info[pos].password << "\n"; test = test
+ 1;
        }
        pos = pos + 1;
    } if (test == 0)
        cout << "No Matches Found\n";
}
```

```

void readfile(Information info[])
{
    ifstream in;
    in.open("/home/118/database1.txt");
    if (in.fail())
    {
        cout << "Cannot read the file\n";
        exit(1);
    }
    int pos = 0;
    while (true)
    {
        in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
            >> info[pos].day >> info[pos].month >> info[pos].year
            >> info[pos].password; if (in.fail())
            break;
        pos = pos + 1;
    }
    in.close();
    pos = 0;
    while (pos < 10)
    {
        cout << info[pos].firstname << " " << info[pos].lastname
            << " " << info[pos].ssn << " " << info[pos].day
            << " " << info[pos].month << " " << info[pos].year
            << " " << info[pos].year << " " << info[pos].password << "\n";
        pos = pos + 1;
    }
    pos = 990;
    while (pos <= 999)
    {
        cout << info[pos].firstname << " " << info[pos].lastname
            << " " << info[pos].ssn << " " << info[pos].day
            << " " << info[pos].month << " " << info[pos].year
            << " " << info[pos].year << " " << info[pos].password << "\n";
        pos = pos + 1;
    }
}

int main()
{
    Information info[1000];
    readfile(info);
    string name;
    cin >> name;
    search(name, info); }

```

```
int main()
{
    Information info[1000];
    readfile(info);
    string name;
    cin >> name; search(name, info);
}

sma457@rabbit:~ % CC lab11.cpp
sma457@rabbit:~ % a.out
Betty Lamprey 101440980 10 8 1957 1957 XaLSkKlm
Pinky Fellows 103980155 24 6 1927 1927 v6ysEjYJ
Oliver Kringle 106100993 14 3 1947 1947 ndmBakcV
Larry Brown 108690448 5 1 1948 1948 un64bkdW
Adolf Davies 111180874 20 10 1964 1964 i0Pto0DI
Bella Napster 112208747 27 2 1971 1971 K09emy0z
Marianne Stone 112550407 16 7 1973 1973 zcZzk1ke
Apu Mitchell 113070570 4 5 1957 1957 ce9bnaYv
Jilly Aston 113220519 23 10 1940 1940 OrM9Sey0
Matilda Vincent 114680858 7 6 1967 1967 2nFeLvBa
Cora Wallace 992360644 4 6 1938 1938 BoIpALi0
Fido Billingsley 993100862 2 8 1961 1961 DiktK56M
Darrell Condor 993460662 25 3 1966 1966 axMuzEh0
Norbert Cooke 994630286 7 11 1938 1938 CqEKk84b
Sigmund Jameson 995870375 30 9 1952 1952 mVpLuXBx
Beatrice Abbot 998880533 7 2 1941 1941 SLfEEbeD
Curtis Croydon 999180994 11 1 1967 1967 wQZVl4LN
Gary Reese 999190938 19 6 1978 1978 leVvcPvm
Ebola Lester 999240047 15 2 1937 1937 B8XvVoH0
Devon Paulson 999430041 19 5 1967 1967 XMgqLkT8
Jillian
Jillian Zilog 178940513 18 1 1949 aqPoZn7E
Jillian Avilla 817380788 6 8 1940 jlc7cHUW
sma457@rabbit:~ %
```

### 3. Find the Oldest

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <string>

using namespace std;

struct Information
{
    string firstname, lastname, password;
    int ssn, day, month, year;
};

void swap(Information info[], int first, int pos)
{
    Information x = info[0];
    info[0] = info[pos];
    info[pos] = x;
}

void old(Information name[], int length)
{
    Information oldest = name[0];
```

```

int i = 0;
int pos = 0;
while (i < length)
{
    if (oldest.year > name[i].year)
    {
        oldest = name[i];
        pos = i;
    }
    if (oldest.year == name[i].year && oldest.month > name[i].month)
    {
        oldest = name[i];
        pos = i;
    }
    if (oldest.year == name[i].year && oldest.month == name[i].month && oldest.day >
        name[i].day)
    {
        oldest = name[i];
        pos = i;
    }
    i = i + 1;
}
swap(name, 0, pos);
cout << oldest.firstname << " " << oldest.lastname << " " << oldest.ssn << ' '
<< oldest.day << " " << oldest.month << " " << oldest.year << " "
<< oldest.password << " " << "\n";
}

```

```

void readfile(Information info[])
{
    ifstream in;
    in.open("/home/118/database1.txt");
    if (in.fail())
    {
        cout << "Cannot read the file\n";
        exit(1);
    }

    const int N = 1000;
    info[N].firstname = " ";
    info[N].lastname = " ";
    info[N].password = " ";
    info[N].ssn = 0;
    info[N].day = 00;
    info[N].month = 00;
    info[N].year = 0000;

    int pos = 0;

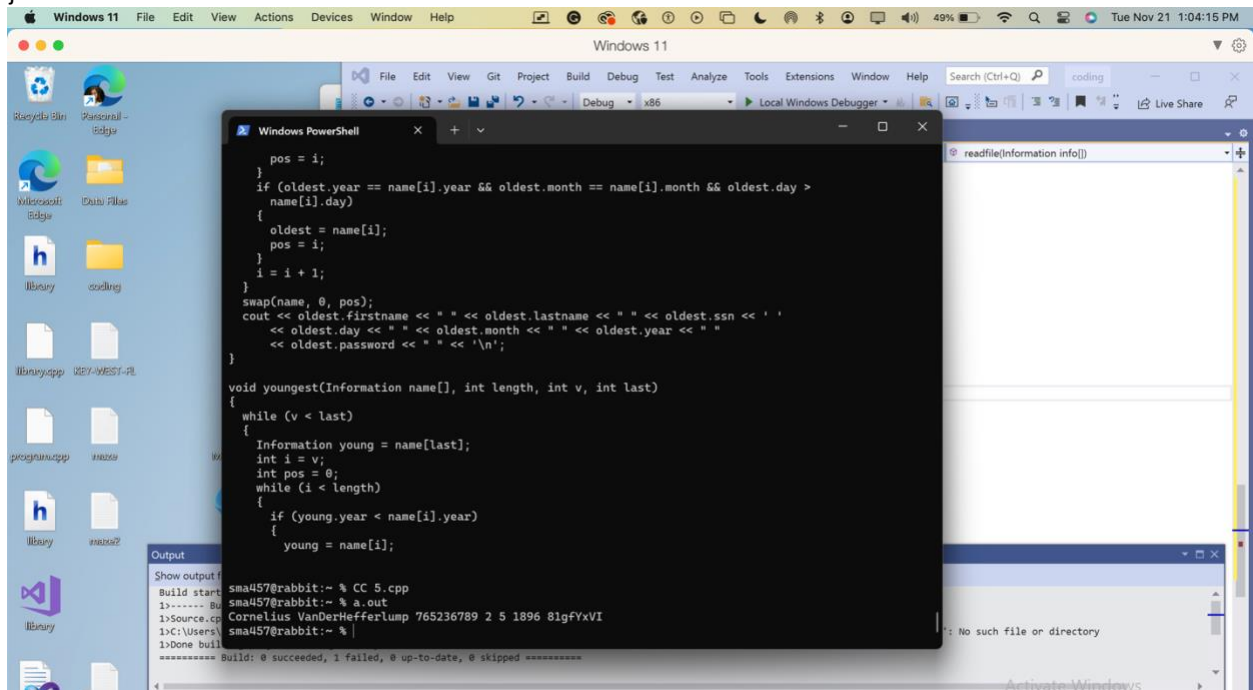
```

```

while (true)
{
    in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
    >> info[pos].day >> info[pos].month >> info[pos].year
    >> info[pos].password;
    if (in.fail())
        break;
    pos = pos + 1;
}
in.close();
old(info, N);
}

int main()
{
    Information info[1001];
    readfile(info);
}

```



#### 4. Promote the Youngest

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <string>

using namespace std;

struct Information
{
    string firstname, lastname, password;
    int ssn, day, month, year;
};

void swap(Information info[], int first, int pos)
{
    Information x = info[0];
    info[0] = info[pos];
    info[pos] = x;
}

void old(Information name[], int length)
{
    Information oldest = name[0];
    int i = 0;
    int pos = 0;
    while (i < length)
    {
        if (oldest.year > name[i].year)
        {
            oldest = name[i];
            pos = i;
        }
        if (oldest.year == name[i].year && oldest.month > name[i].month)
        {
            oldest = name[i];
            pos = i;
        }
        if (oldest.year == name[i].year && oldest.month == name[i].month && oldest.day >
            name[i].day)
        {
            oldest = name[i];
            pos = i;
        }
        i = i + 1;
    }
}
```



```

swap(name, 0, pos);
cout << oldest.firstname << " " << oldest.lastname << " " << oldest.ssn << ' '
    << oldest.day << " " << oldest.month << " " << oldest.year << " "
    << oldest.password << " " << '\n';
}

void youngest(Information name[], int length)
{
    Information young = name[0];
    int i = 0;
    int pos = 0;
    while (i < length)
    {
        if (young.year < name[i].year)
        {
            young = name[i];
            pos = i;
        }
        if (young.year == name[i].year && young.month < name[i].month)
        {
            young = name[i];
            pos = i;
        }
        if (young.year == name[i].year && young.month == name[i].month && young.day <
            name[i].day)
        {
            young = name[i];
            pos = i;
        }
        i = i + 1;
    }
    swap(name, 0, pos);
    cout << young.firstname << " " << young.lastname << " " << young.ssn << ' '
        << young.day << " " << young.month << " " << young.year << " "
        << young.password << " " << '\n';
}

void readfile(Information info[])
{
    ifstream in;
    in.open("/home/118/database1.txt");
    if (in.fail())
    {
        cout << "Cannot read the file\n";
        exit(1);
    }

    const int N = 1000;
    info[N].firstname = " ";

```

```

info[N].lastname = " ";
info[N].password = " ";
info[N].ssn = 0;
info[N].day = 00;
info[N].month = 00;
info[N].year = 0000;

```

```

int pos = 0;
while (true)
{
    in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
    >> info[pos].day >> info[pos].month >> info[pos].year
    >> info[pos].password;
    if (in.fail())
        break;
    pos = pos + 1;
}

```

```

in.close();
old(info, N);

```

```

}

```

```

int main()
{
    Information info[1001];
    readfile(info);
}

```

The screenshot shows a Windows 11 desktop environment. In the background, a Visual Studio Code window is open, displaying a C++ program. The program includes a function to read information into an array and a function to find the oldest person based on year, month, and day. In the foreground, a Windows PowerShell terminal window is open, showing the execution of the program. The output displays the details of the oldest person found: Hal Morris, born on 12/12/1984, with an SSN of Hx10X3k8.

```

int i = 0;
int pos = 0;
while (i < length)
{
    if (oldest.year < name[i].year)
    {
        oldest = name[i];
        pos = i;
    }
    if (oldest.year == name[i].year && oldest.month < name[i].month)
    {
        oldest = name[i];
        pos = i;
    }
    if (oldest.year == name[i].year && oldest.month == name[i].month && oldest.day <
        name[i].day)
    {
        oldest = name[i];
        pos = i;
    }
    i = i + 1;
}
swap(name, 0, pos);
cout << oldest.firstname << " " << oldest.lastname << " " << oldest.ssn << " " << "\n";

```

```

sma457@rabbit:~$ g++ CC_5.cpp
sma457@rabbit:~$ ./a.out
Hal Morris 179190468 26 12 1984 Hx10X3k8
sma457@rabbit:~$

```

## 5. Now Promote the Second Youngest

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <string>

using namespace std;

struct Information
{
    string firstname, lastname, password;
    int ssn, day, month, year;
};

void swap(Information info[], int first, int pos)
{
    Information x = info[first];
    info[first] = info[pos];
    info[pos] = x;
}

void old(Information name[], int length)
{
    Information oldest = name[0];
    int i = 0;
    int pos = 0;
    while (i < length)
    {
        if (oldest.year > name[i].year)
        {
            oldest = name[i];
            pos = i;
        }
        if (oldest.year == name[i].year && oldest.month > name[i].month)
        {
            oldest = name[i];
            pos = i;
        }
        if (oldest.year == name[i].year && oldest.month == name[i].month && oldest.day >
            name[i].day)
        {
            oldest = name[i];
            pos = i;
        }
        i = i + 1;
    }
    swap(name, 0, pos);
    cout << oldest.firstname << " " << oldest.lastname << " " << oldest.ssn << " "
        << oldest.day << " " << oldest.month << " " << oldest.year << " "
        << oldest.password << " " << '\n';
}
```

```

void youngest(Information name[], int length, int v, int last)
{
    while (v < 2)
    {
        Information young = name[last];
        int i = v;
        int pos = 0;
        while (i < length)
        {
            if (young.year < name[i].year)
            {
                young = name[i];
                pos = i;
            }
            if (young.year == name[i].year && young.month < name[i].month)
            {
                young = name[i];
                pos = i;
            }
            if (young.year == name[i].year && young.month == name[i].month && young.day <
                name[i].day)
            {
                young = name[i];
                pos = i;
            }
            i = i + 1;
        }
        swap(name, v, pos);
        v = v + 1;
        cout << young.firstname << " " << young.lastname << " " << young.ssn << ' '
            << young.day << " " << young.month << " " << young.year << " "
            << young.password << " " << "\n";
    }
}

```

```

void readfile(Information info[])
{
    ifstream in;
    in.open("/home/118/database1.txt");
    if (in.fail())
    {
        cout << "Cannot read the file\n";
        exit(1);
    }
}

```

```

const int N = 1000;

```

```

info[N].firstname = " ";
info[N].lastname = " ";
info[N].password = " ";
info[N].ssn = 0;
info[N].day = 00;
info[N].month = 00;
info[N].year = 0000;

int pos = 0;
while (true)
{
    in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
        >> info[pos].day >> info[pos].month >> info[pos].year
        >> info[pos].password;
    if (in.fail())
        break;
    pos = pos + 1;
}

in.close();
youngest(info, N, 0, N);
}

```

```

int main()
{
    Information info[1001];
    readfile(info);
}

```

```

i = i + 1;
}
swap(name, 0, pos);
cout << oldest.firstname << " " << oldest.lastname << " " << oldest.ssn << " "
    << oldest.day << " " << oldest.month << " " << oldest.year << " "
    << oldest.password << " " << "\n";
}

void youngest(Information name[], int length, int v, int last)
{
    while (v < 2)
    {
        Information young = name[last];
        int i = v;
        int pos = 0;
        while (i < length)
        {
            if (young.year < name[i].year)
            {
                young = name[i];
                pos = i;
            }
            if (young.year == name[i].year && young.month < name[i].month)
            {
                young = name[i];
                pos = i;
            }
        }
        last = pos;
        v++;
    }
}

sma457@rabbit:~ % CC 11.cpp
sma457@rabbit:~ % a.out
Hal Morris 179190468 26 12 1984 Hx10X3k8
Imelda Jones 490220429 3 12 1984 M18ZaWk7
sma457@rabbit:~ %

```

## 6. More of the Same.

```

#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <string>

using namespace std;

struct Information
{
    string firstname, lastname, password;
    int ssn, day, month, year;
};

void swap(Information info[], int first, int pos)
{
    Information x = info[first];
    info[first] = info[pos];
    info[pos] = x;
}

void old(Information name[], int length)
{
    Information oldest = name[0];
    int i = 0;
    int pos = 0;
    while (i < length)
    {
        if (oldest.year > name[i].year)
        {
            oldest = name[i];
            pos = i;
        }
        if (oldest.year == name[i].year && oldest.month > name[i].month)
        {
            oldest = name[i];
            pos = i;
        }
        if (oldest.year == name[i].year && oldest.month == name[i].month && oldest.day >
            name[i].day)
        {
            oldest = name[i];
            pos = i;
        }
        i = i + 1;
    }
    swap(name, 0, pos);
}

```

```

    cout << oldest.firstname << " " << oldest.lastname << " " << oldest.ssn << ' '
        << oldest.day << " " << oldest.month << " " << oldest.year << " "
        << oldest.password << " " << '\n';
}

void youngest(Information name[], int length, int v, int last, int a)
{
    while (v < a)
    {
        Information young = name[last];
        int i = v;
        int pos = 0;
        while (i < length)
        {
            if (young.year < name[i].year)
            {
                young = name[i];
                pos = i;
            }
            if (young.year == name[i].year && young.month < name[i].month)
            {
                young = name[i];
                pos = i;
            }
            if (young.year == name[i].year && young.month == name[i].month && young.day <
                name[i].day)
            {
                young = name[i];
                pos = i;
            }
            i = i + 1;
        }
        swap(name, v, pos);
        v = v + 1;
        cout << young.firstname << " " << young.lastname << " " << young.ssn << ' '
            << young.day << " " << young.month << " " << young.year << " "
            << young.password << " " << '\n';
    }
}

void readfile(Information info[])
{
    ifstream in;
    in.open("/home/118/database1.txt");
    if (in.fail())
    {
        cout << "Cannot read the file\n";
        exit(1);
    }
}

```

```

}

const int N = 1000;
info[N].firstname = " ";
info[N].lastname = " ";
info[N].password = " ";
info[N].ssn = 0;
info[N].day = 00;
info[N].month = 00;
info[N].year = 0000;

int pos = 0;
while (true)
{
    in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
        >> info[pos].day >> info[pos].month >> info[pos].year
        >> info[pos].password;
    if (in.fail())
        break;
    pos = pos + 1;
}

in.close();
youngest(info, N, 0, N, 3);
}

int main()
{
    Information info[1001];
    readfile(info);
}

```

```

Information young = name[last];
int i = v;
int pos = 0;
while (i < length)
{
    if (young.year < name[i].year)
    {
        young = name[i];
        pos = i;
    }
    if (young.year == name[i].year && young.month < name[i].month)
    {
        young = name[i];
        pos = i;
    }
    if (young.year == name[i].year && young.month == name[i].month && young.day <
        name[i].day)
    {
        young = name[i];
        pos = i;
    }
    i = i + 1;
}

sma457@rabbit:~$ CC 13.cpp
sma457@rabbit:~$ a.out
Hal Morris 179198468 26 12 1984 Hx10X3k8
Imelda Jones 498228429 3 12 1984 Ml8ZaMK7
Anabel Mosley 372198483 2 11 1984 wkwUSrR
sma457@rabbit:~$

```

## 7. The Ultimate Demand.



```

#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <string>

using namespace std;

struct Information
{
    string firstname, lastname, password;
    int ssn, day, month, year;
};

void swap(Information info[], int first, int pos)
{
    Information x = info[first];
    info[first] = info[pos];
    info[pos] = x;
}

void youngest(Information name[], int length, int v, int last)
{
    while (v < last)
    {
        Information young = name[last];
        int i = v;
        int pos = 0;
        while (i < length)
        {
            if (young.year < name[i].year)
            {
                young = name[i];
                pos = i;
            }
            if (young.year == name[i].year && young.month < name[i].month)
            {
                young = name[i];
                pos = i;
            }
            if (young.year == name[i].year && young.month == name[i].month &&
young.day <
                name[i].day)
            {
                young = name[i];
                pos = i;
            }
            i = i + 1;
        }
    }
}

```

```

        }
        swap(name, v, pos);
        v = v + 1;
    }
}

void readfile(Information info[])

{
    ifstream in;
    in.open("/home/118/database100.txt");
    if (in.fail())
    {
        cout << "Cannot read the file\n";
        exit(1);
    }

f
    const int N = 100000;
    info[N].firstname = " ";
    info[N].lastname = " ";
    info[N].password = " ";
    info[N].ssn = 0;
    info[N].day = 00;
    info[N].month = 00;
    info[N].year = 0000;

    int pos = 0;
    while (true)
    {
        in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
            >> info[pos].day >> info[pos].month >> info[pos].year
            >> info[pos].password;
        if (in.fail())
            break;
        pos = pos + 1;
    }

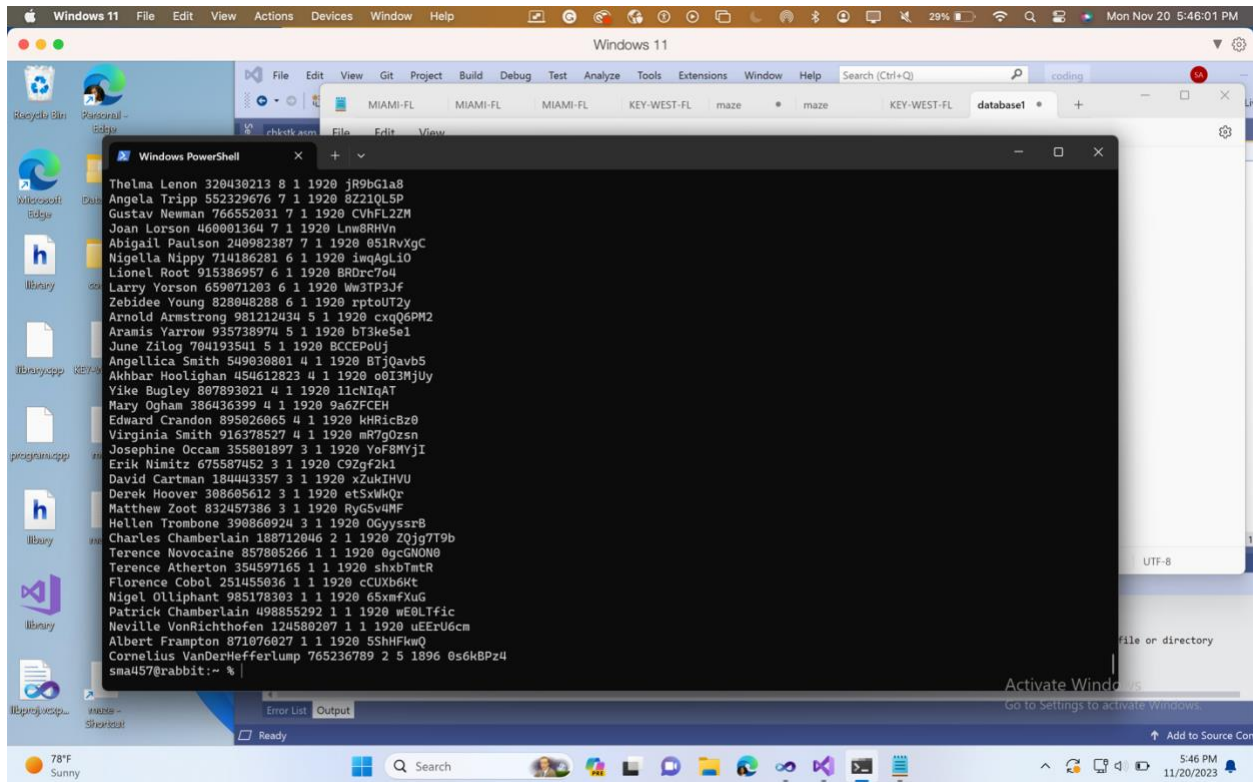
    in.close();
    youngest(info, N, 0, N);
    int i = 0;
    while (i < N)
    {
        cout << info[i].firstname << " " << info[i].lastname << " " << info[i].ssn << " " <<
info[i].day
            << " " << info[i].month << " " << info[i].year << " " << info[i].password << "\n";
        i = i + 1;
    }
}

```

```

int main()
{
    Information info[100001];
    readfile(info);
}

```



## 8. Sorting the File.

```

#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <string>

```

```
using namespace std;
```

```

struct Information
{
    string firstname, lastname, password;
    int ssn, day, month, year;
};

```

```

void swap(Information info[], int first, int pos)
{

```

```

        Information x = info[first];
        info[first] = info[pos];
        info[pos] = x;
    }

void youngest(Information name[], int length, int v, int last)
{
    while (v < last)
    {
        Information young = name[last];
        int i = v;
        int pos = 0;
        while (i < length)
        {
            if (young.year < name[i].year)
            {
                young = name[i];
                pos = i;
            }
            if (young.year == name[i].year && young.month < name[i].month)
            {
                young = name[i];
                pos = i;
            }
            if (young.year == name[i].year && young.month == name[i].month &&
young.day <
                name[i].day)
            {
                young = name[i];
                pos = i;
            }
            i = i + 1;
        }
        swap(name, v, pos);
        v = v + 1;
    }
}

void readfile(Information info[])
{
    ifstream in;
    in.open("/home/118/database100.txt");
    if (in.fail())
    {
        cout << "Cannot read the file\n";
        exit(1);
    }
    ofstream young("young.txt");

```

```

const int N = 100000;
info[N].firstname = " ";
info[N].lastname = " ";
info[N].password = " ";
info[N].ssn = 0;
info[N].day = 00;
info[N].month = 00;
info[N].year = 0000;

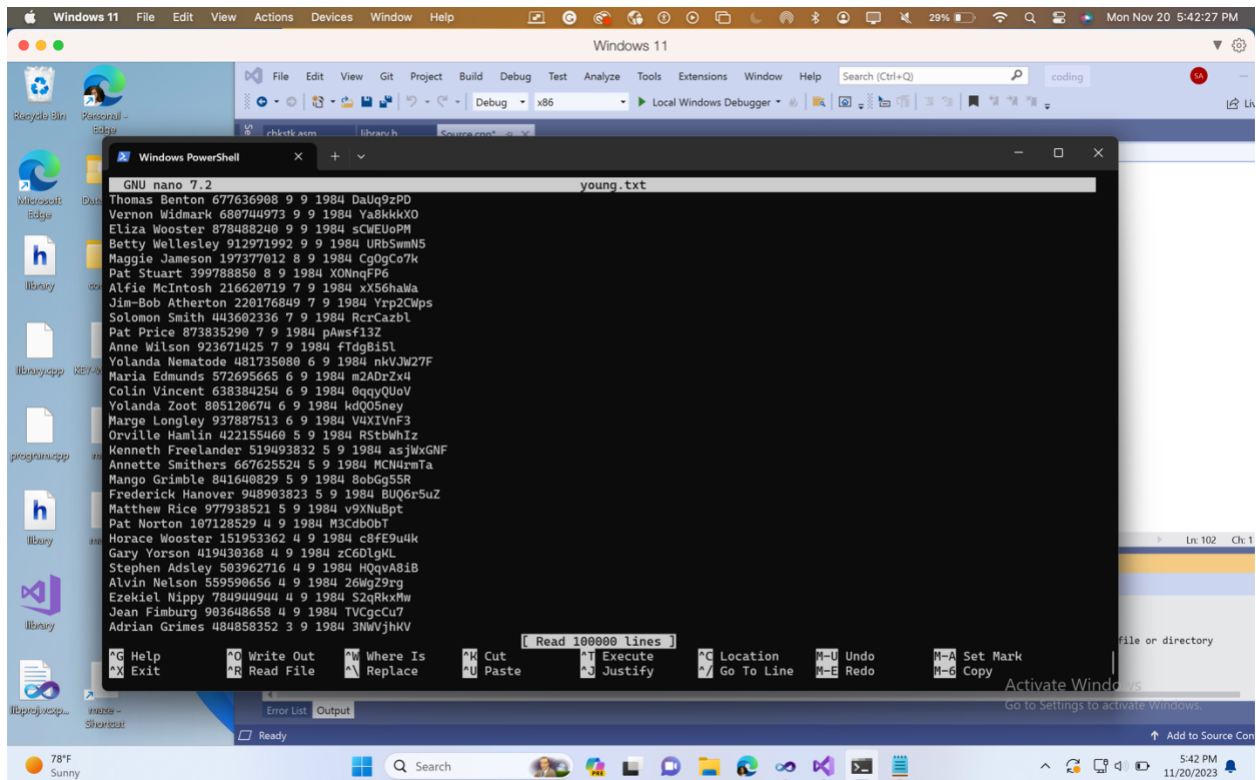
int pos = 0;
while (true)
{
    in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
        >> info[pos].day >> info[pos].month >> info[pos].year
        >> info[pos].password;
    if (in.fail())
        break;
    pos = pos + 1;
}

in.close();
youngest(info, N, 0, N);
int i = 0;
while (i < N)
{
    young << info[i].firstname << " " << info[i].lastname << " " << info[i].ssn << " " <<
info[i].day
        << " " << info[i].month << " " << info[i].year << " " << info[i].password << "\n";
    i = i + 1;
}

}

int main()
{
    Information info[100001];
    readfile(info); }

```



## 9. How Fast Is It?

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <string>
#include <time.h>
#include <sys/resource.h>
```

using namespace std;

```
struct Information
```

```
{
    string firstname, lastname, password;
    int ssn, day, month, year;
};
```

```
double get_cpu_time()
```

```
{
    rusage ruse;
    getrusage(RUSAGE_SELF, &ruse);
    return ruse.ru_utime.tv_sec + ruse.ru_utime.tv_usec / 1000000.0 +
           ruse.ru_stime.tv_sec + ruse.ru_stime.tv_usec / 1000000.0;
}
```

```
void swap(Information info[], int first, int pos)
```

```
{  
    Information x = info[first];  
    info[first] = info[pos];  
    info[pos] = x;  
}
```

```
void youngest(Information name[], int length, int v, int last)
```

```
{  
    double x = get_cpu_time();  
    while (v < last)  
    {  
        Information young = name[last];  
        int i = v;  
        int pos = 0;  
        while (i < length)  
        {  
            if (young.year < name[i].year)  
            {  
                young = name[i];  
                pos = i;  
            }  
            if (young.year == name[i].year && young.month < name[i].month)  
            {  
                young = name[i];  
                pos = i;  
            }  
            if (young.year == name[i].year && young.month == name[i].month &&  
young.day <  
                name[i].day)  
            {  
                young = name[i];  
                pos = i;  
            }  
            i = i + 1;  
        }  
        swap(name, v, pos);  
        v = v + 1;  
    }  
    cout << "CPU TIME: " << x << "\n";  
}
```

```
void readfile(Information info[])
```

```
{  
    ifstream in;  
    in.open("/home/118/database100.txt");  
}
```

```

if (in.fail())
{
    cout << "Cannot read the file\n";
    exit(1);
}
ofstream young("young.txt");

const int N = 100000;
info[N].firstname = " ";
info[N].lastname = " ";
info[N].password = " ";
info[N].ssn = 0;
info[N].day = 00;
info[N].month = 00;
info[N].year = 0000;

int pos = 0;
while (true)
{
    in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
        >> info[pos].day >> info[pos].month >> info[pos].year
        >> info[pos].password;
    if (in.fail())
        break;
    pos = pos + 1;
}

in.close();
youngest(info, N, 0, N);
int i = 0;
while (i < N)
{
    young << info[i].firstname << " " << info[i].lastname << " " << info[i].ssn << " " <<
info[i].day
        << " " << info[i].month << " " << info[i].year << " " << info[i].password << "\n";
    i = i + 1;
}
}

int main()
{
    Information info[100001];
    readfile(info);
}

```



