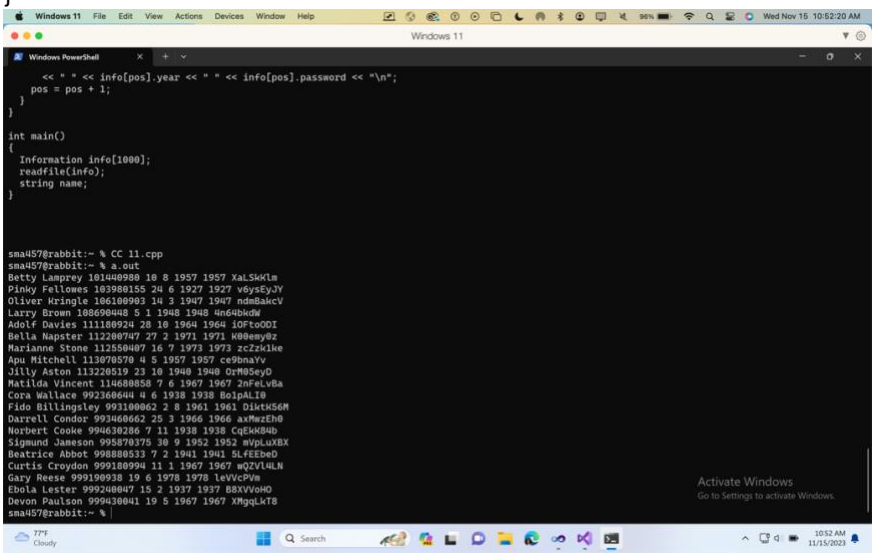
Lab #10

ECE 118 – Section R/RC
Lab on Wednesday at 5:05
Sloan Atkins

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)
	{
```

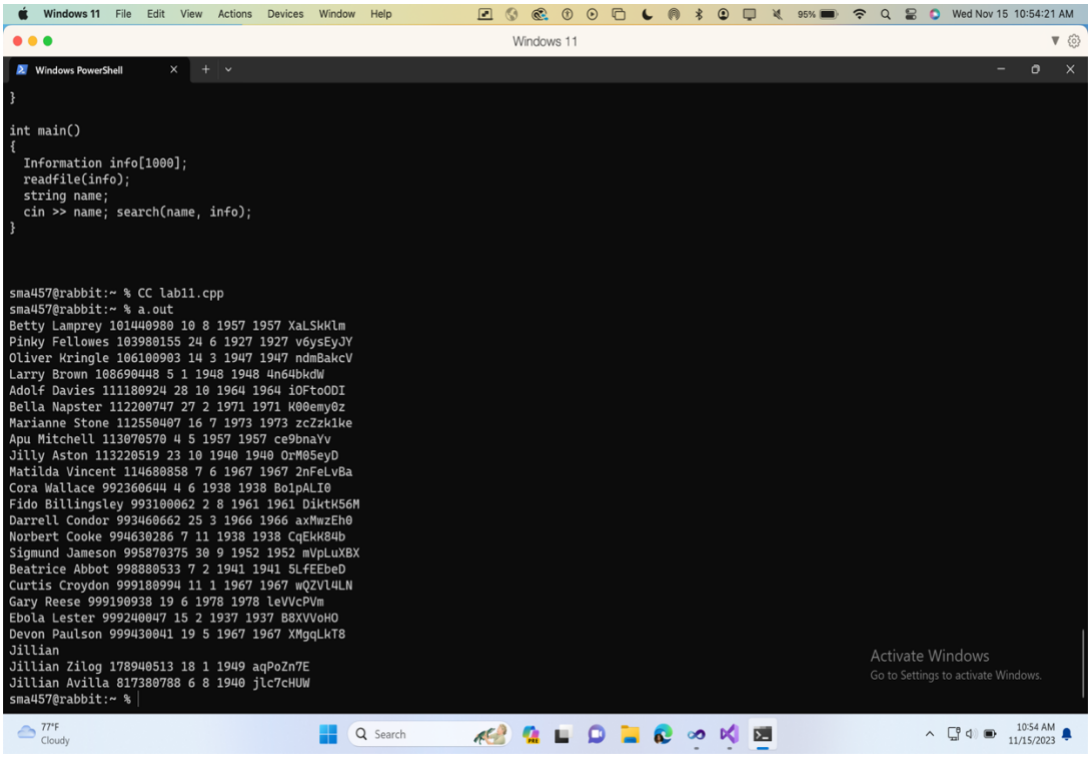
```
            in >> info[pos].firstname >> info[pos].lastname >> info[pos].ssn
                  >> info[pos].day >> info[pos].month >> info[pos].year
                  >> info[pos].password; if (in.fail())
                  break;
            pos = pos + 1;
      }
      in.close();
      pos = 0;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



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); }
```

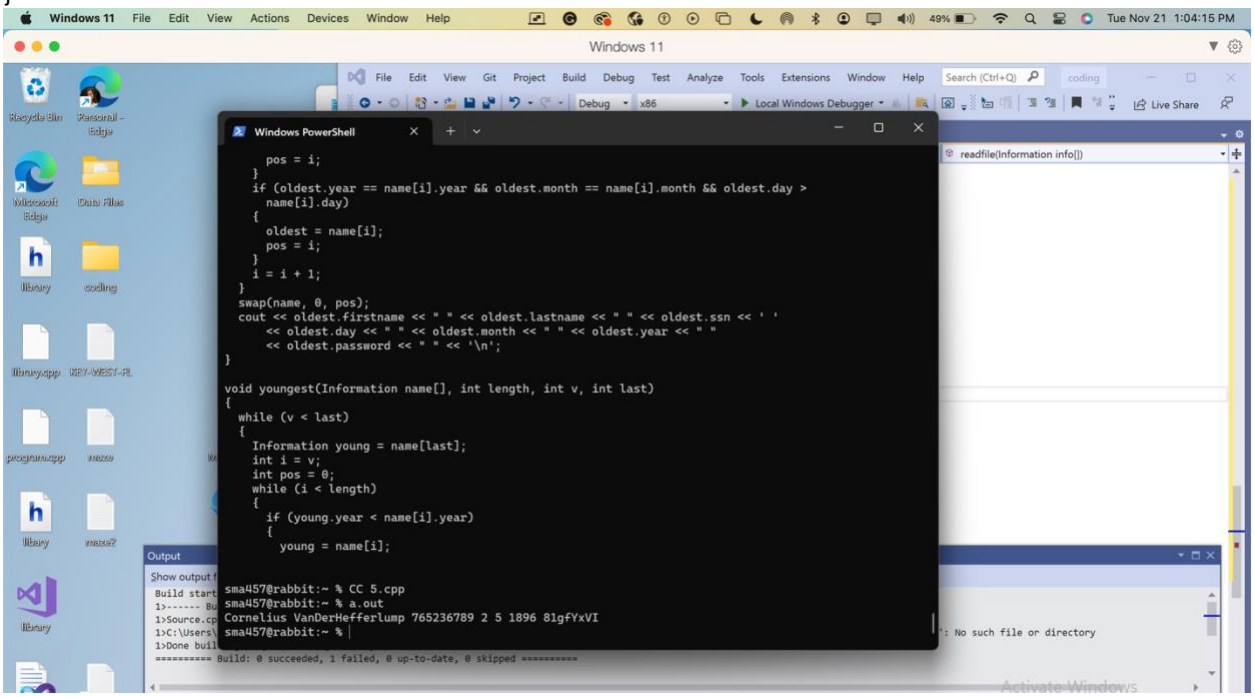


3. Find the Oldest

```
#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);
}
```

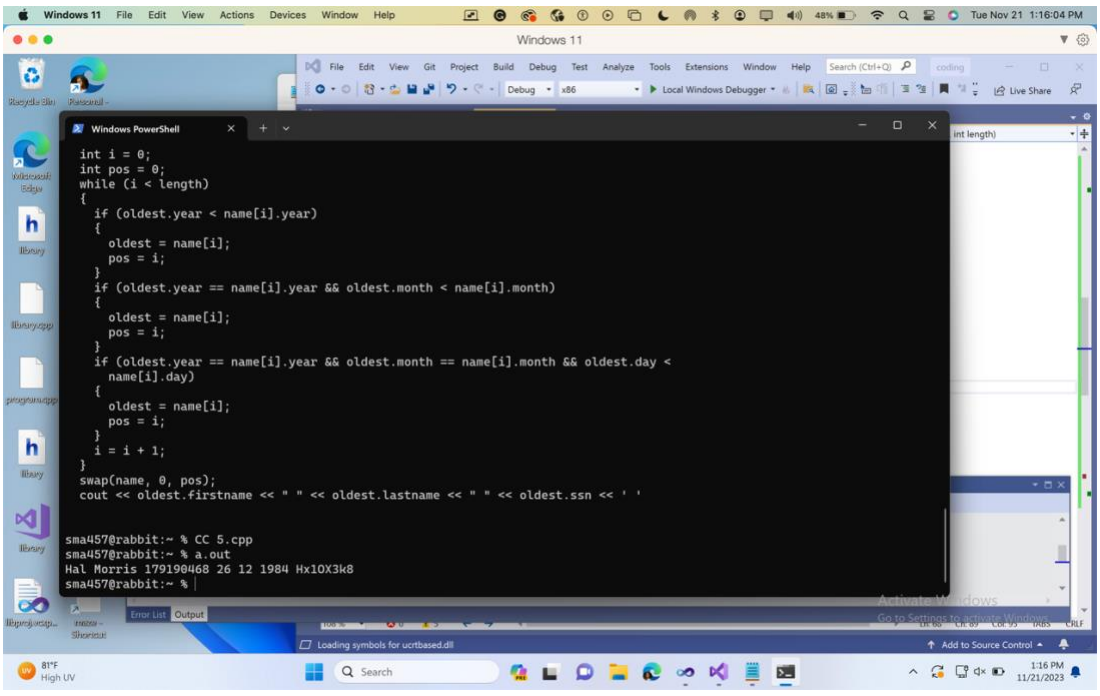


4. Promote the Youngest

```
#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);
}
```

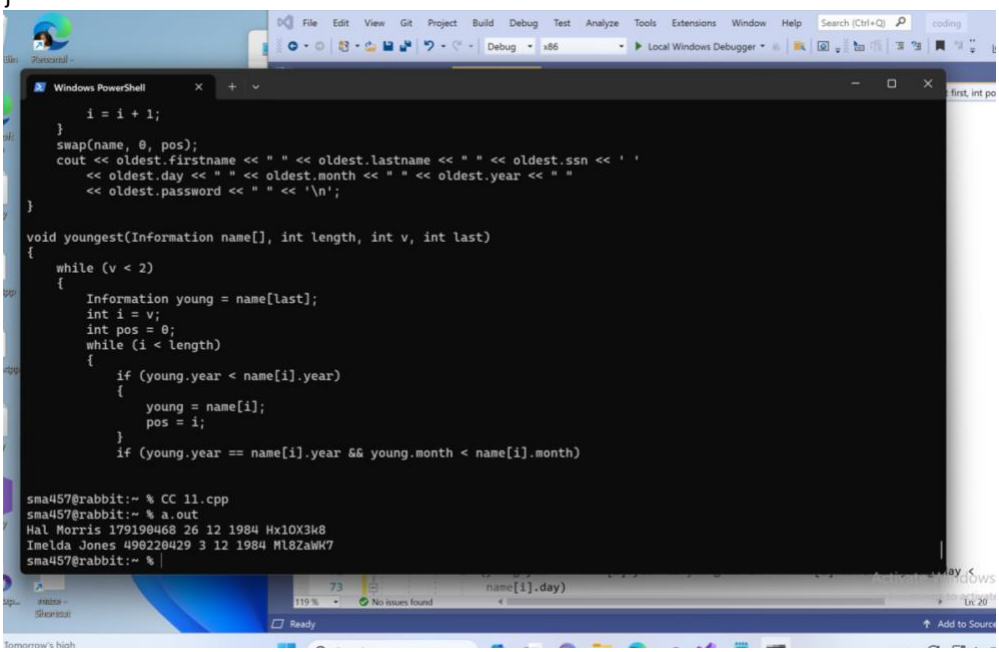


5. **Now Promote the Second Youngest**

```
#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);
}
```

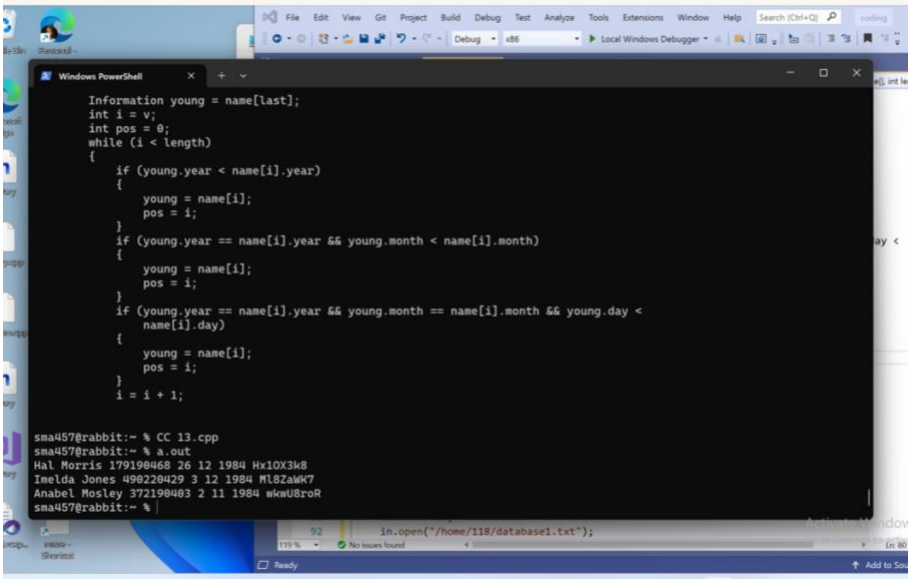


6. More of the Same.

```
#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);
}
```

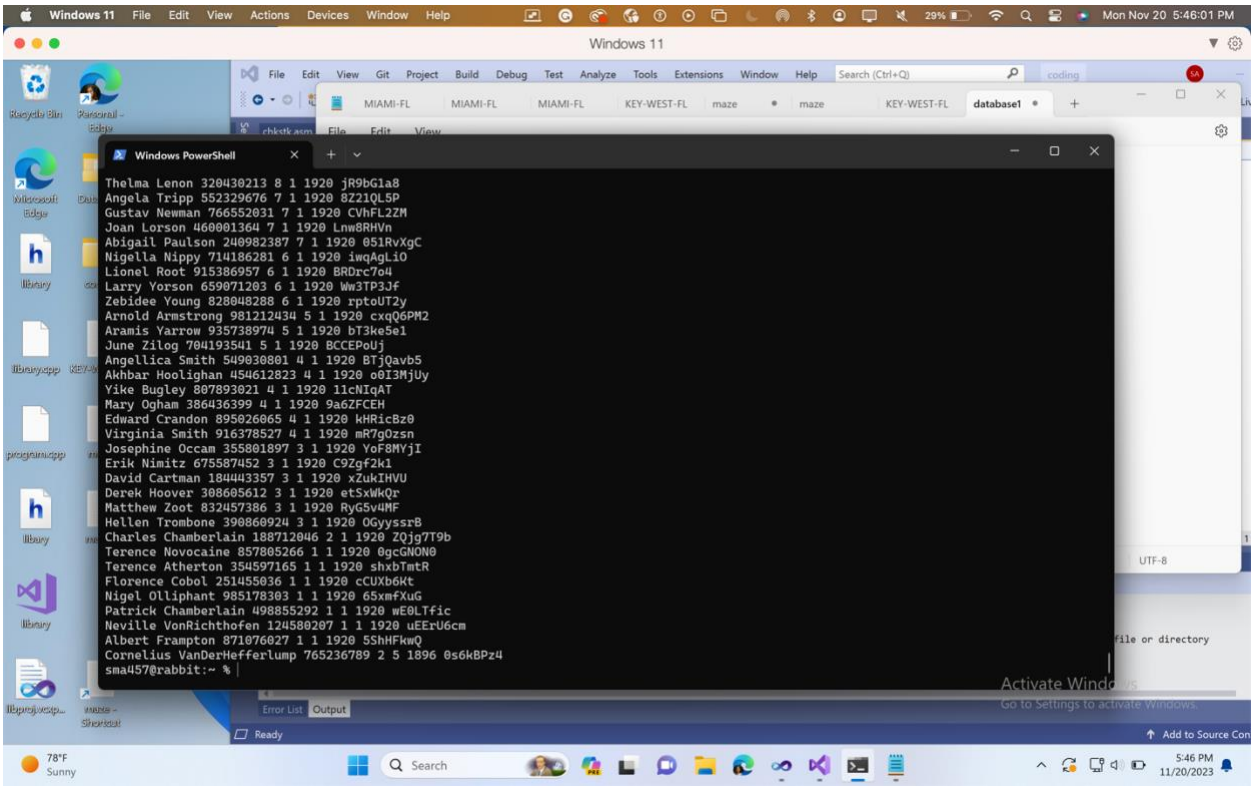


7. The Ultimate Demand.

```
#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);
}
```

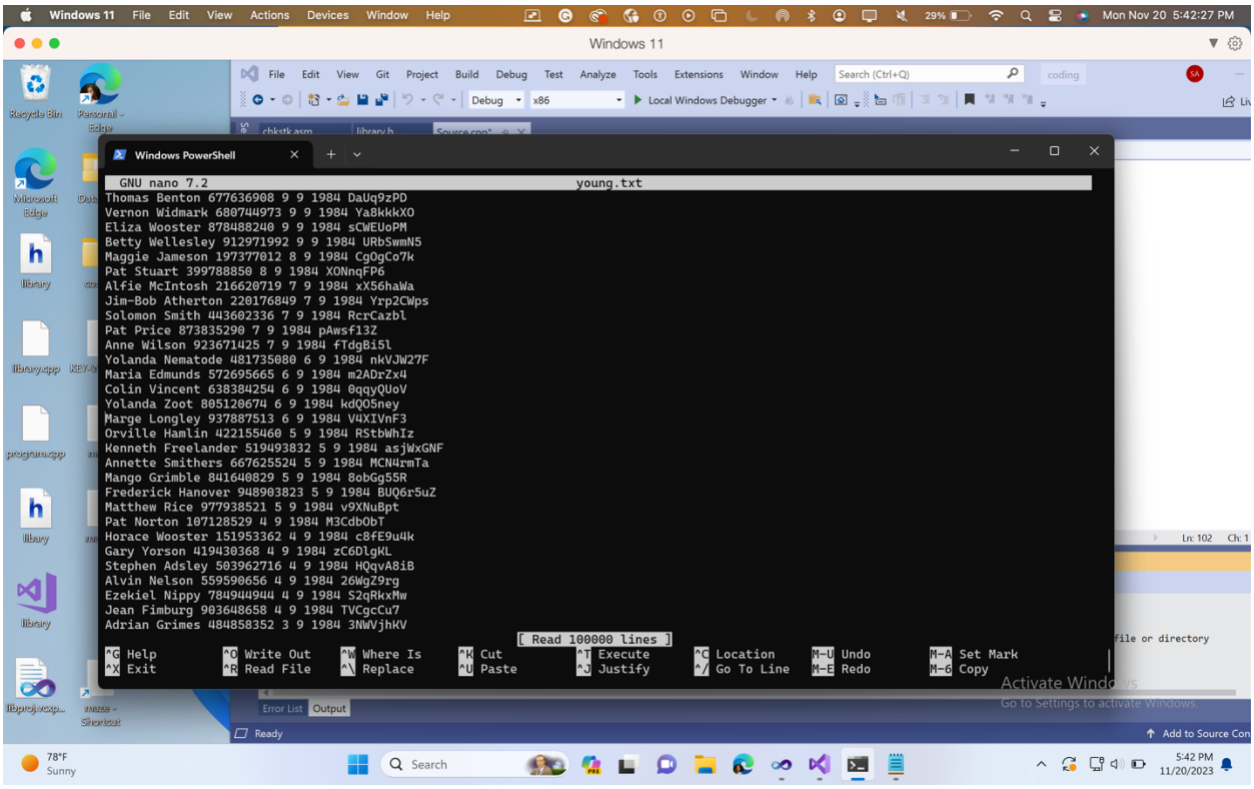


8. Sorting the File.

```
#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); }
```

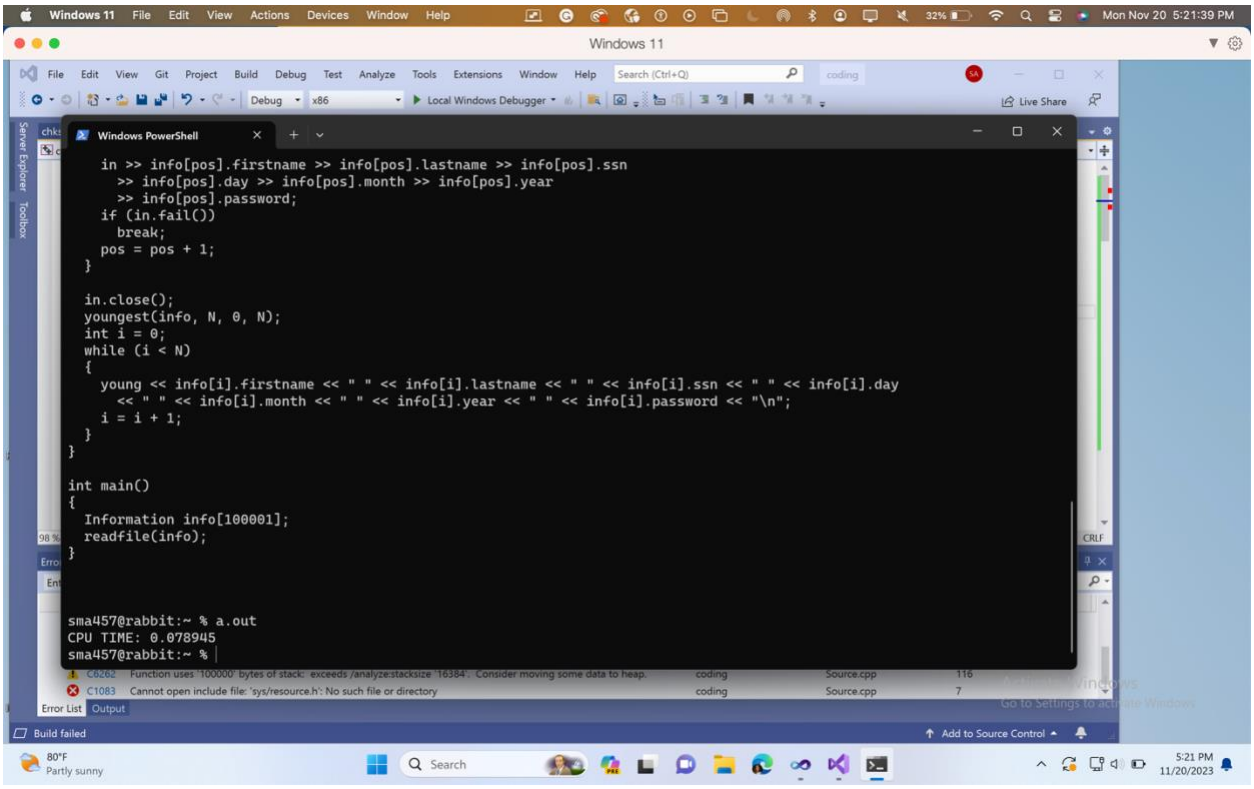


9. How Fast Is It?

```
#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);
}
```


Windows 11
Windows PowerShell
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);
}
sma457@rabbit:~ % a.out
CPU TIME: 0.078945
sma457@rabbit:~ %
Build failed
Add to Source Control
80°F
Partly sunny
Search
5:21 PM
11/20/2023