Lab Assignment 0
DE1-SoCLinux Getting Started

Levi Kaplan
kaplan.l@northeastern.edu

Submit Date: 9/14/20
Due Date: 9/14/20

Abstract

In this lab, we learned the basics of C++ and gained experience with coding, compiling, and running code on the DE1-SoC board. We transferred files between out computer and the board and experimented with sorting algorithms and I/O in C++.

Introduction

In this lab, we perform various introductory tasks such as setting up the DE1-SoC board. We create files on the board, transfer files between our computer and the board, and write small programs that interface with the board. Additionally, we copy, modify, and improve code written in earlier portions of the lab. Taken together, this lab represents an introduction to the expectations and curriculum of the class.

Software and Hardware Used

Hardware

- DE1-SoC board
- Ethernet cable

Software

- Atom text editor
- Online C++ compiler (https://www.onlinegdb.com/online_c++_compiler)
- ssh and scp
- nano and vim

Lab Steps

Lab 0.0

1. We first configured the board
2. We next configured the board's IP address
3. We then set our Mac's IP address to match the board's
4. We then ssh'ed into the board
5. In a separate terminal, we can now use scp to securely transfer files

Lab 0.1

1. We created a directory named lab0 in the home folder of the board
2. We used nano to create a file named hello.cpp
3. We wrote a *Hello World* program in this file
4. We used g++ to compile this
5. We executed the program
6. We saved this file in our personal computer

7. We used pwd and ls -l to show the path to the directory, and took screenshot
8. We replaced std::cout with just cout and observed that the output was the same

Lab 0.2

1. We used nano to create the C++ file hello2.cpp
2. We wrote the second *Hello World* program
3. We compiled the program
4. We executed the program
5. We saved the file in our personal computer
6. We used pwd and ls -l to show the path to the directory, and took screenshot
7. We deleted *using namespace std;* from the program and observed that we got an error that cout wasn't defined

Lab 0.3

1. We used nano to create a lab03.cpp file
2. We created the randomarray array to hold 10 integer values
3. We used generated a random number, and saved it into randomarray[0]
4. We generated 10 random numbers and save them to randomarray
5. We displayed the values in randomarray
6. We sorted the array
7. We displayed the now sorted values in randomarray
8. We took a screenshot of the code
9. We used pwd and ls -l to show the path of the directory and ran the program, taking a screenshot of the output and the path

Lab 0.4

1. We copied the content of lab03.cpp into lab04.cpp
2. We outputted the address of randomarray and the other variables used in the program
3. We screenshotted the code used in lab04.cpp
4. We used pwd and ls -l to show the path of the directory and ran the program, taking a screenshot of the output and the path

Lab 0.5

1. We copied the content of lab03.cpp into array.cpp
2. We wrote the PrintArray function
3. We wrote the RandomArray function
4. We wrote the SortArray function
5. We removed redundant code
6. We transferred this file to the board, compiled, and ran it
7. We screenshotted the written code in array.cpp
8. We used pwd and ls -l to show the path of the directory and ran the program, taking a screenshot of the output and the path

Lab 0.6

1. We copied array.cpp into string.cpp
2. We modified main to work with strings instead of integers
3. We modified the methods in array.cpp to take in strings instead of integers
4. We screenshotted the C++ code in string.cpp
5. We screenshotted the commands to compile the source and run it, as well as the path and current directory

Lab Discussion

Lab 0.0

1. We first connected power, mini usb, and ethernet to the board
2. We then connected ethernet and usb to computer
3. We turned the board on
4. We next configured the board's IP address
    a. determine where the board is by running ls /dev/tty.*
    b. screen into the board by running screen /dev/tty.*board* 115200
    c. set the ip by running ifconfig eth0 192.168.1.1
5. We next wentto system preferences → networks
    a. in the usb-lan menu, set Configure to Manually
    b. IP Address: 192.168.0.1
    c. Subnet Mask: 255.255.255.0
6. We ran ssh root@192.168.0.123
    a. Now, we were ssh'ed in to the board
7. In a separate terminal, we can now use scp to securely transfer files

   a. scp *file* root@192.168.0.123:/home/root

Lab 0.1

1. We created a directory named lab0 in the home folder of the board
   a. mkdir lab0
2. We used nano to create a file named hello.cpp
   a. nano hello.cpp
   b. We wrote a *Hello World* program in this file
3. We used g++ to compile this
   a. g++ -o runhello hello.cpp
4. We executed the program
   a. ./runhello
5. We saved this file in our personal computer
   a. using ssh in our home terminal, we used scp to transfer the file from the board at root@192.168.0.123 to our personal directory
6. We showed the path to the directory, and took screenshot
   a. pwd and ls -l
7. We replaced std::cout with just cout and observed that the output was the same

Lab 0.2

1. We used nano to create the C++ file hello2.cpp
   a. nano hello2.ccp
2. We wrote the second *Hello World* program
3. We compiled the program with g++
   a. g++ -o helloworld2 hello2.cpp
4. We executed the program
   a. ./helloworld2
5. We saved the file in our personal computer
   a. using ssh in our home terminal, we used scp to transfer the file from the board at root@192.168.0.123 to our personal directory
6. We used pwd and ls -l to show the path to the directory, and took screenshot
7. We deleted *using namespace std;* from the program and observed that we got an error that cout wasn't defined

Lab 0.3

1. We used nano to create a lab03.cpp file
   a. nano lab03.cpp
2. We created the randomarray array to hold 10 integer values
3. We compiled the program
4. We used srand(time(NULL)) to generate a random number, and saved this into randomarray[0]
5. We compiled the program
6. We used a for loop to generate 10 random numbers and save them to randomarray
7. We compiled the program
8. We used a for loop to display the values in randomarray
9. We compiled the program
10. We used selection sort to sort the array
11. We compiled the program
12. We used a for loop to display the now sorted values in randomarray
13. We compiled the program
14. We took a screenshot of the code
15. We used pwd and ls -l to show the path of the directory and ran the program, taking a screenshot of the output and the path

Lab 0.4

1. We used the cp command to copy the content of lab03.cpp into lab04.cpp
2. We used &*variable name* to output the address of randomarray and the other variables used in the program
3. We screenshotted the code used in lab04.cpp
4. We used pwd and ls -l to show the path of the directory and ran the program, taking a screenshot of the output and the path

Lab 0.5

1. We used the cp command to copy the content of lab03.cpp into array.cpp
2. We wrote the PrintArray function
3. We wrote the RandomArray function
4. We wrote the SortArray function

5. We modified Main to remove the now redundant code that has been abstracted into separate functions, and now merely calls those functions
6. We transferred this file to the board, compiled, and ran it
7. We screenshotted the written code in array.cpp
8. We used pwd and ls -l to show the path of the directory and ran the program, taking a screenshot of the output and the path

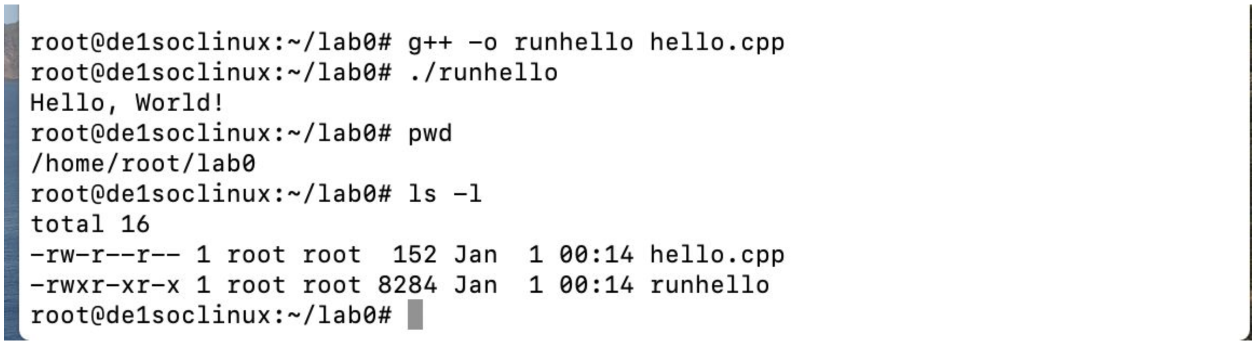
Lab 0.6

1. We used the cp command to copy array.cpp into string.cpp
2. We made 10 different string variables in the main method
3. We assigned the input to those string variables using cin
4. We created an array of string values called stringarray
5. We added the 10 different strings into stringarray
6. We modified the methods in array.cpp to take in strings instead of integers
7. We screenshotted the C++ code in string.cpp
8. We screenshotted the commands to compile the source and run it, as well as the path and current directory

Results

lab 0.1

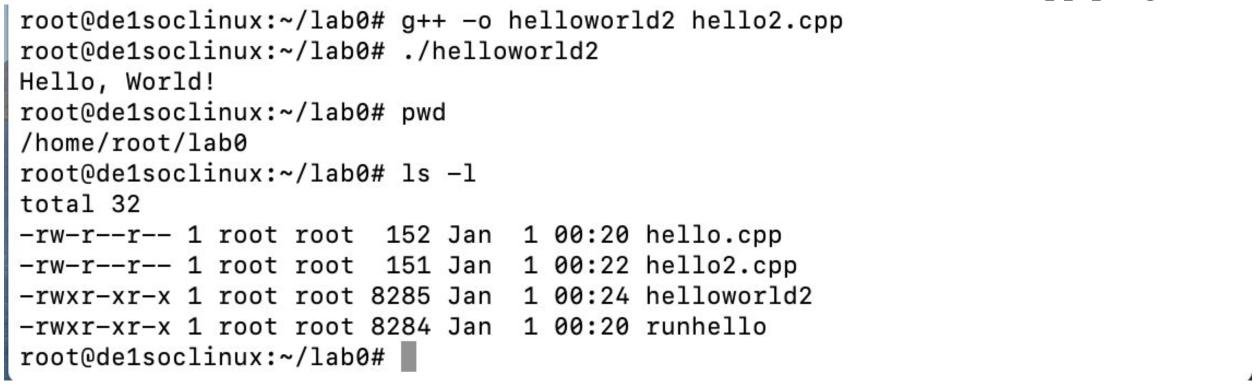
My hello.cpp program worked as intended. It successfully printed "Hello, World!" to the console.



```
root@de1soclinux:~/lab0# g++ -o runhello hello.cpp
root@de1soclinux:~/lab0# ./runhello
Hello, World!
root@de1soclinux:~/lab0# pwd
/home/root/lab0
root@de1soclinux:~/lab0# ls -l
total 16
-rw-r--r-- 1 root root  152 Jan  1 00:14 hello.cpp
-rwxr-xr-x 1 root root 8284 Jan  1 00:14 runhello
root@de1soclinux:~/lab0# 
```

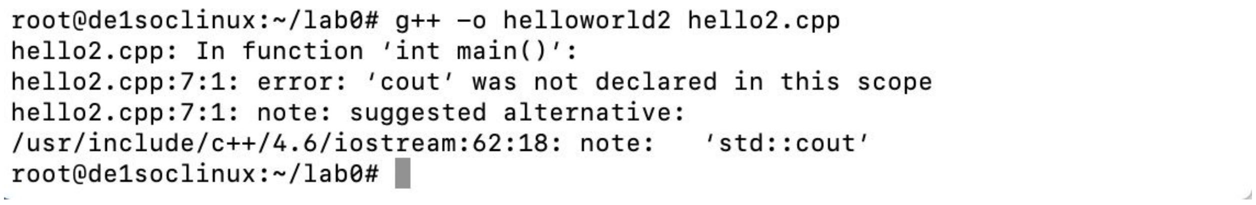
Lab 0.2

The hello2.cpp program was identical to the hello.cpp program except this program omitted the std:: before cout. The behavior was identical to the hello.cpp program.



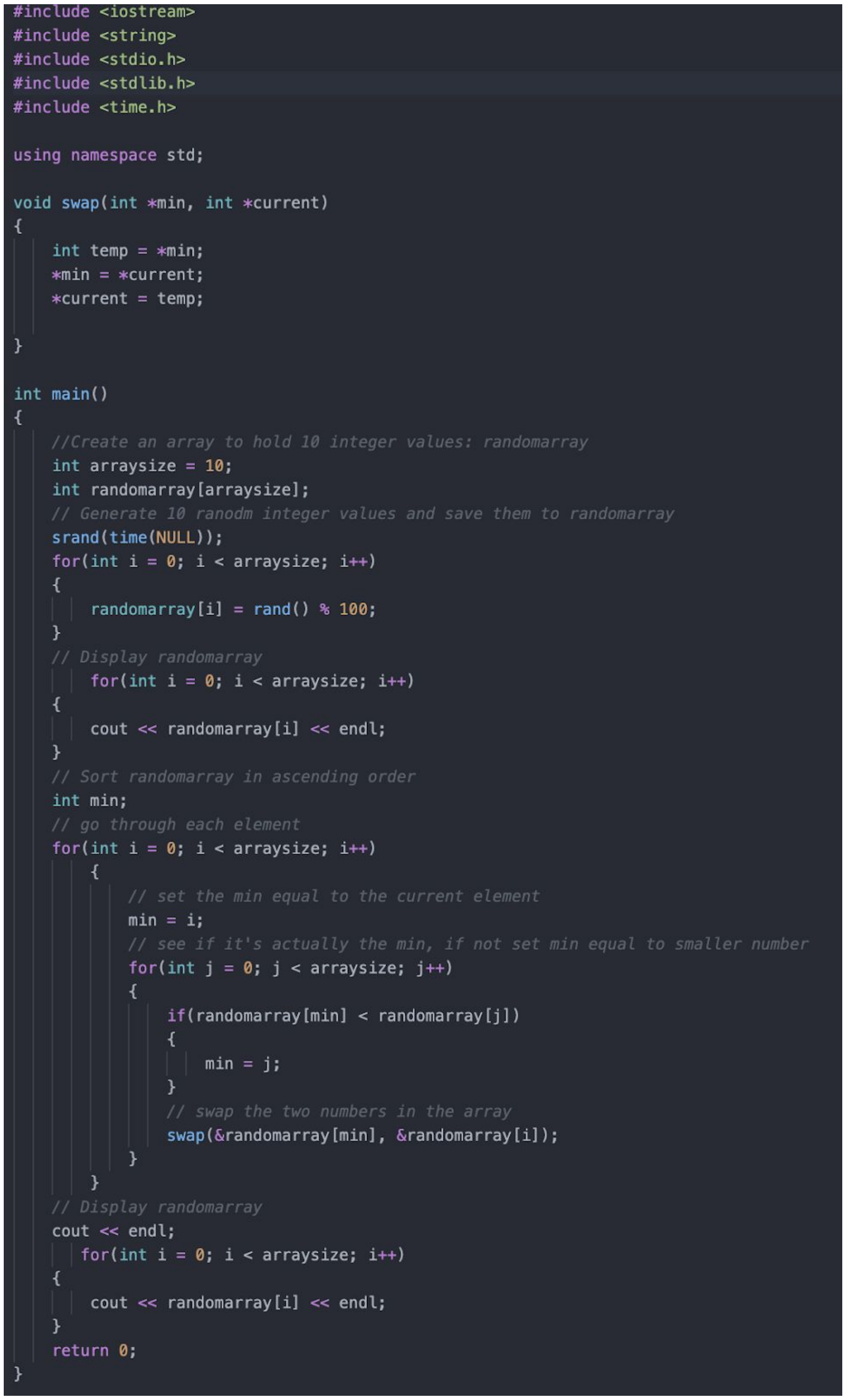
```
root@de1soclinux:~/lab0# g++ -o helloworld2 hello2.cpp
root@de1soclinux:~/lab0# ./helloworld2
Hello, World!
root@de1soclinux:~/lab0# pwd
/home/root/lab0
root@de1soclinux:~/lab0# ls -l
total 32
-rw-r--r-- 1 root root  152 Jan  1 00:20 hello.cpp
-rw-r--r-- 1 root root  151 Jan  1 00:22 hello2.cpp
-rwxr-xr-x 1 root root 8285 Jan  1 00:24 helloworld2
-rwxr-xr-x 1 root root 8284 Jan  1 00:20 runhello
root@de1soclinux:~/lab0#
```

When we omitted ***using namespace std;*** this is the error we got. It’s saying that ‘cout’ wasn’t declared in the scope, because standard in and out aren’t defined when this line is omitted.



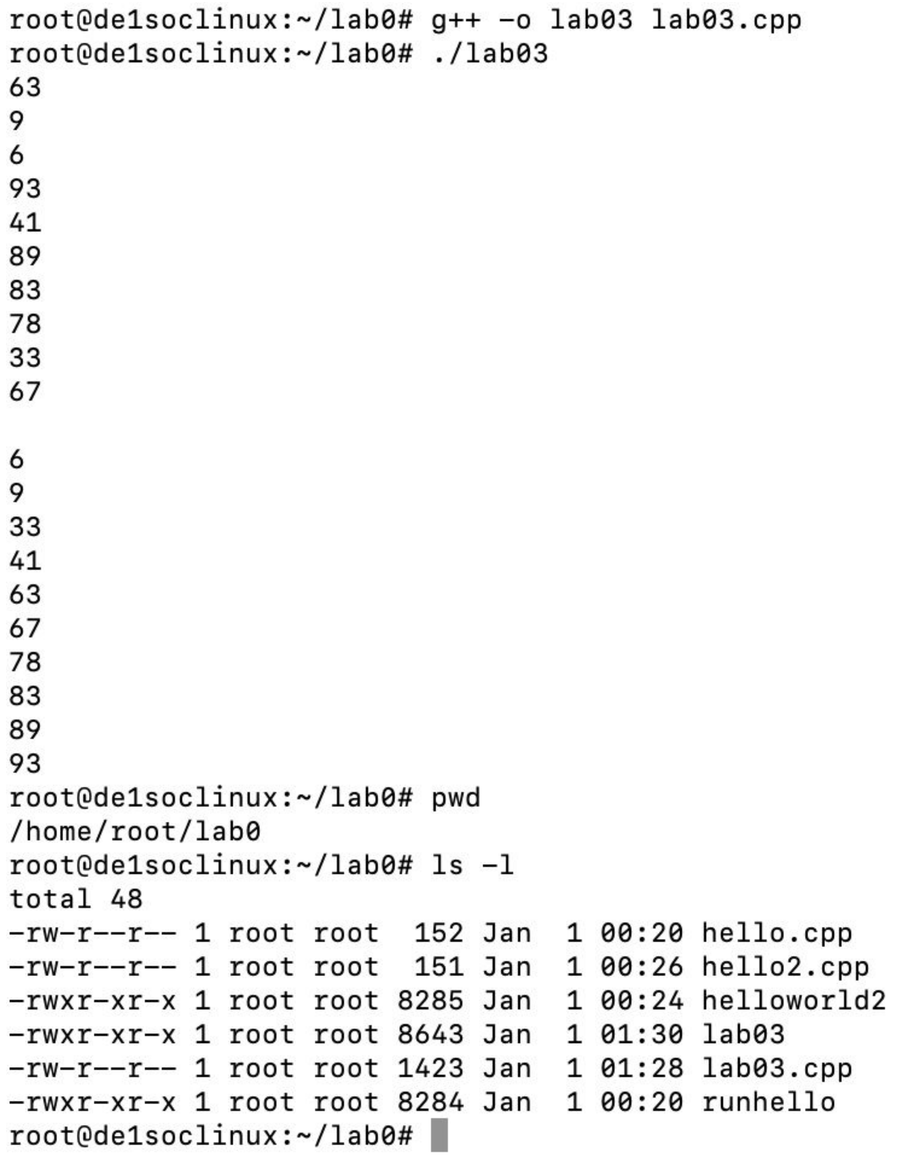
```
root@de1soclinux:~/lab0# g++ -o helloworld2 hello2.cpp
hello2.cpp: In function ‘int main()’:
hello2.cpp:7:1: error: ‘cout’ was not declared in this scope
hello2.cpp:7:1: note: suggested alternative:
/usr/include/c++/4.6/iostream:62:18: note:   ‘std::cout’
root@de1soclinux:~/lab0#
```

Lab 0.3



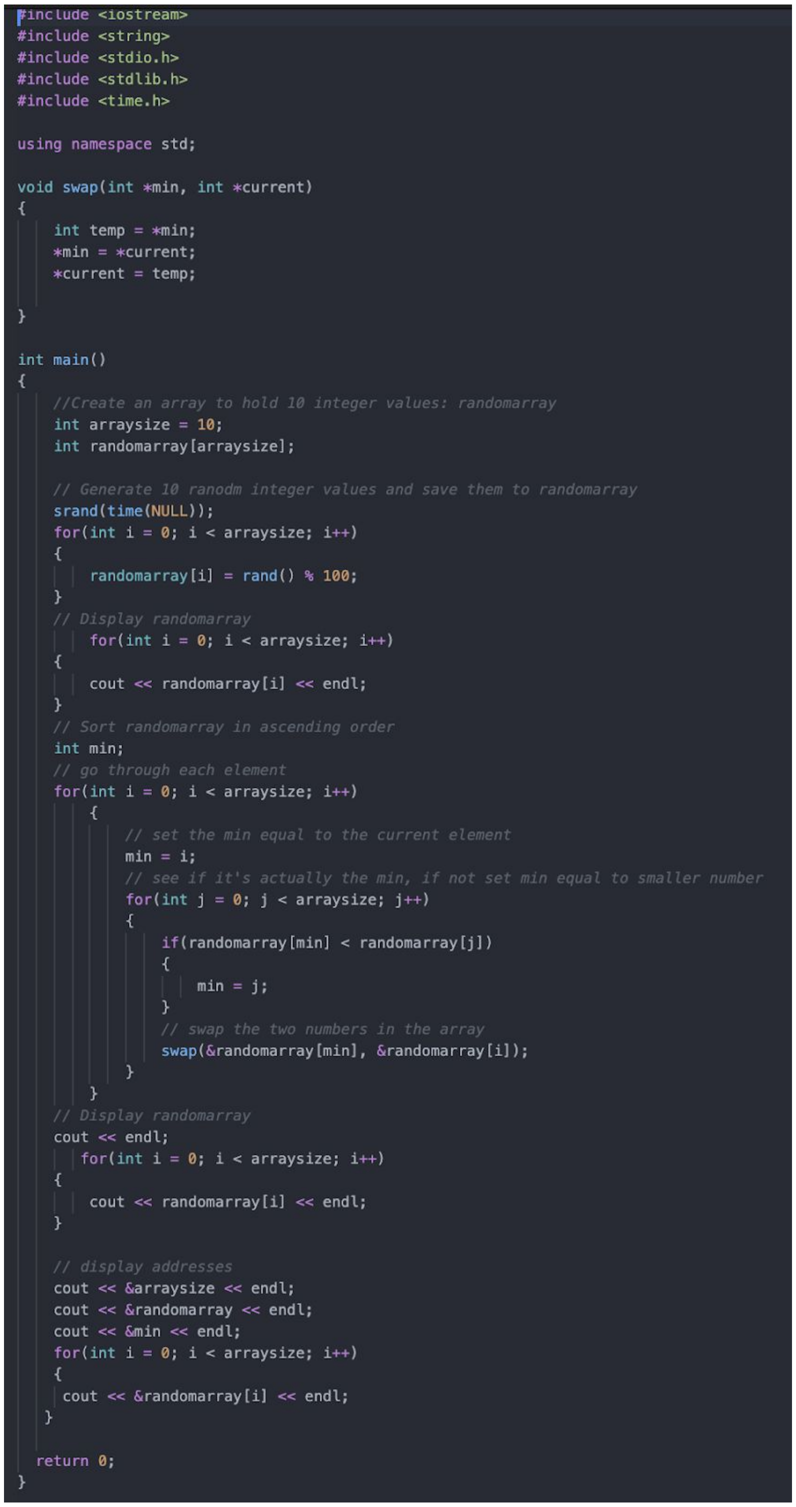
```
#include <iostream>
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

using namespace std;

void swap(int *min, int *current)
{
    int temp = *min;
    *min = *current;
    *current = temp;

}

int main()
{
    //Create an array to hold 10 integer values: randomarray
    int arraysize = 10;
    int randomarray[arraysize];
    // Generate 10 ranodm integer values and save them to randomarray
    srand(time(NULL));
    for(int i = 0; i < arraysize; i++)
    {
        randomarray[i] = rand() % 100;
    }
    // Display randomarray
        for(int i = 0; i < arraysize; i++)
    {
        cout << randomarray[i] << endl;
    }
    // Sort randomarray in ascending order
    int min;
    // go through each element
    for(int i = 0; i < arraysize; i++)
        {
            // set the min equal to the current element
            min = i;
            // see if it's actually the min, if not set min equal to smaller number
            for(int j = 0; j < arraysize; j++)
            {
                if(randomarray[min] < randomarray[j])
                {
                    min = j;
                }
                // swap the two numbers in the array
                swap(&randomarray[min], &randomarray[i]);
            }
        }
    // Display randomarray
    cout << endl;
       for(int i = 0; i < arraysize; i++)
    {
        cout << randomarray[i] << endl;
    }
    return 0;
}
```

Above is the code written in the Atom text editor for this portion of the lab. The screenshot is not of vim or nano because the entire program could not fit in a single screen.
Below is the output of this program. The 10 random numbers are first printed unsorted, then sorted in increasing order, then printed sorted.



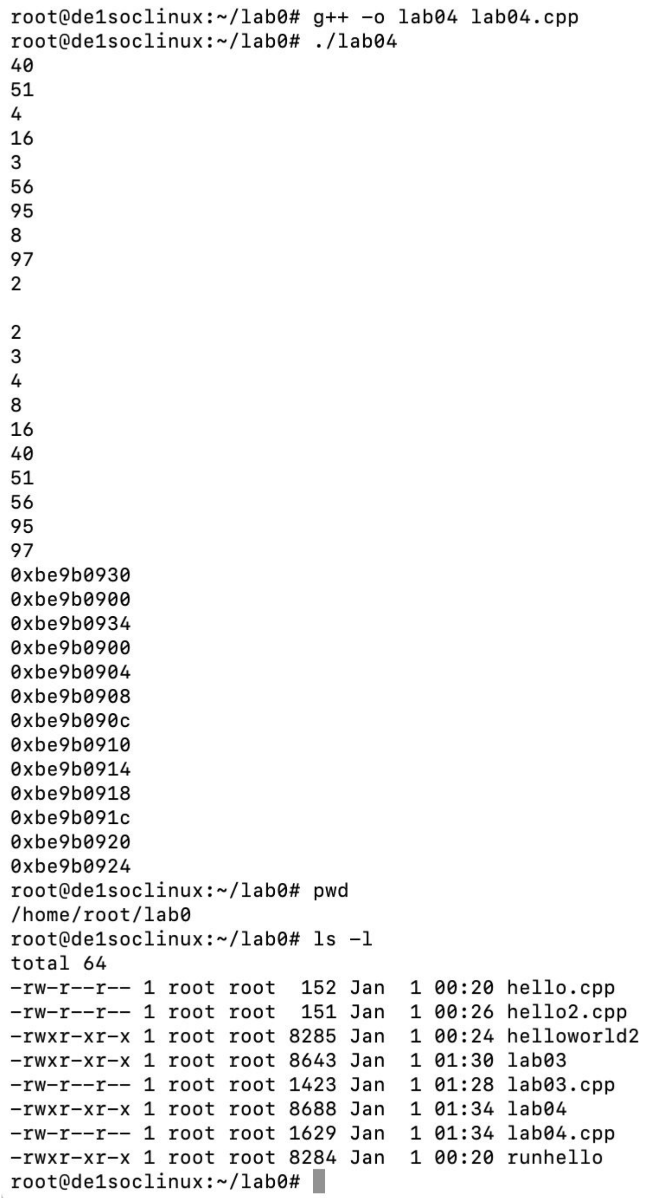
```
root@de1soclinux:~/lab0# g++ -o lab03 lab03.cpp
root@de1soclinux:~/lab0# ./lab03
63
9
6
93
41
89
83
78
33
67

6
9
33
41
63
67
78
83
89
93
root@de1soclinux:~/lab0# pwd
/home/root/lab0
root@de1soclinux:~/lab0# ls -l
total 48
-rw-r--r-- 1 root root  152 Jan  1 00:20 hello.cpp
-rw-r--r-- 1 root root  151 Jan  1 00:26 hello2.cpp
-rwxr-xr-x 1 root root 8285 Jan  1 00:24 helloworld2
-rwxr-xr-x 1 root root 8643 Jan  1 01:30 lab03
-rw-r--r-- 1 root root 1423 Jan  1 01:28 lab03.cpp
-rwxr-xr-x 1 root root 8284 Jan  1 00:20 runhello
root@de1soclinux:~/lab0# 
```

Lab 0.4



```cpp
#include <iostream>
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

using namespace std;

void swap(int *min, int *current)
{
    int temp = *min;
    *min = *current;
    *current = temp;

}

int main()
{
    //Create an array to hold 10 integer values: randomarray
    int arraysize = 10;
    int randomarray[arraysize];

    // Generate 10 ranodm integer values and save them to randomarray
    srand(time(NULL));
    for(int i = 0; i < arraysize; i++)
    {
        randomarray[i] = rand() % 100;
    }
    // Display randomarray
        for(int i = 0; i < arraysize; i++)
    {
        cout << randomarray[i] << endl;
    }
    // Sort randomarray in ascending order
    int min;
    // go through each element
    for(int i = 0; i < arraysize; i++)
        {
            // set the min equal to the current element
            min = i;
            // see if it's actually the min, if not set min equal to smaller number
            for(int j = 0; j < arraysize; j++)
            {
                if(randomarray[min] < randomarray[j])
                {
                    min = j;
                }
                // swap the two numbers in the array
                swap(&randomarray[min], &randomarray[i]);
            }
        }
    // Display randomarray
    cout << endl;
      for(int i = 0; i < arraysize; i++)
    {
        cout << randomarray[i] << endl;
    }

    // display addresses
    cout << &arraysize << endl;
    cout << &randomarray << endl;
    cout << &min << endl;
    for(int i = 0; i < arraysize; i++)
    {
    cout << &randomarray[i] << endl;
  }

  return 0;
}
```

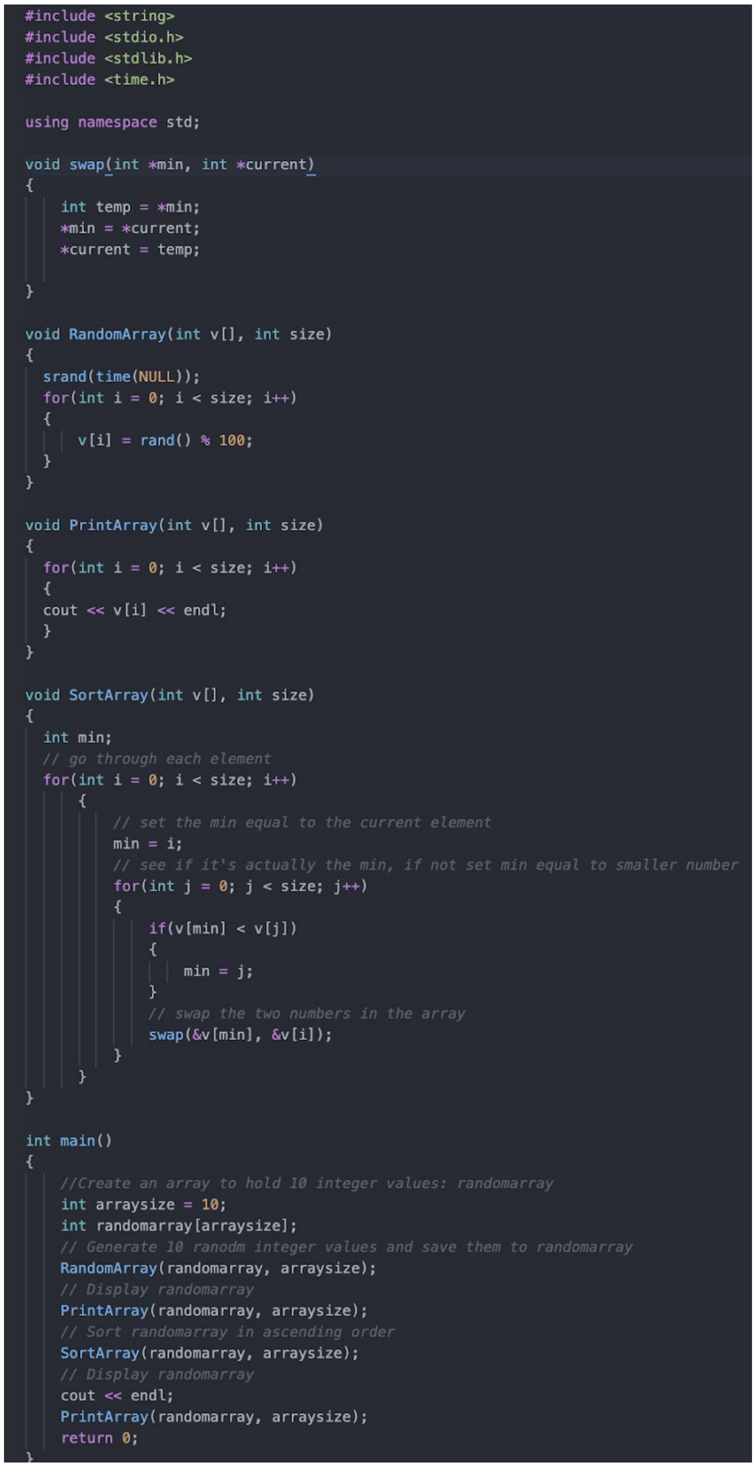
Above is the code for this section. It's very similar to the code for Lab 0.3, except it contains code to print the data addresses for the various variables in the program. Below is the output of this program. One interesting observation I made is that the rearranging of the data does not change the addresses of the numbers in the array. Another interesting observation is that the randomarray itself and the first element of the randomarray have the same address. This could be due to the way arrays are stored in C++.



```
root@de1soclinux:~/lab0# g++ -o lab04 lab04.cpp
root@de1soclinux:~/lab0# ./lab04
40
51
4
16
3
56
95
8
97
2

2
3
4
8
16
40
51
56
95
97
0xbe9b0930
0xbe9b0900
0xbe9b0934
0xbe9b0900
0xbe9b0904
0xbe9b0908
0xbe9b090c
0xbe9b0910
0xbe9b0914
0xbe9b0918
0xbe9b091c
0xbe9b0920
0xbe9b0924
root@de1soclinux:~/lab0# pwd
/home/root/lab0
root@de1soclinux:~/lab0# ls -l
total 64
-rw-r--r-- 1 root root  152 Jan  1 00:20 hello.cpp
-rw-r--r-- 1 root root  151 Jan  1 00:26 hello2.cpp
-rwxr-xr-x 1 root root 8285 Jan  1 00:24 helloworld2
-rwxr-xr-x 1 root root 8643 Jan  1 01:30 lab03
-rw-r--r-- 1 root root 1423 Jan  1 01:28 lab03.cpp
-rwxr-xr-x 1 root root 8688 Jan  1 01:34 lab04
-rw-r--r-- 1 root root 1629 Jan  1 01:34 lab04.cpp
-rwxr-xr-x 1 root root 8284 Jan  1 00:20 runhello
root@de1soclinux:~/lab0# 
```

Variable Name	Memory Address	Memory Values
arraysize	0xbe9b0930	10
randomarray	0xbe9b0900	{ 2, 40, 51, 4, 16, 3, 56, 95, 8, 97, 2 }
min	0xbe9b0934	2
randomarray[0]	0xbe9b0900	40
randomarray[1]	0xbe9b0904	51
randomarray[2]	0xbe9b0908	4
randomarray[3]	0xbe9b090c	16
randomarray[4]	0xbe9b0910	3
randomarray[5]	0xbe9b0914	56
randomarray[6]	0xbe9b0918	95
randomarray[7]	0xbe9b091c	8
randomarray[8]	0xbe9b0920	97
randomarray[9]	0xbe9b0924	2

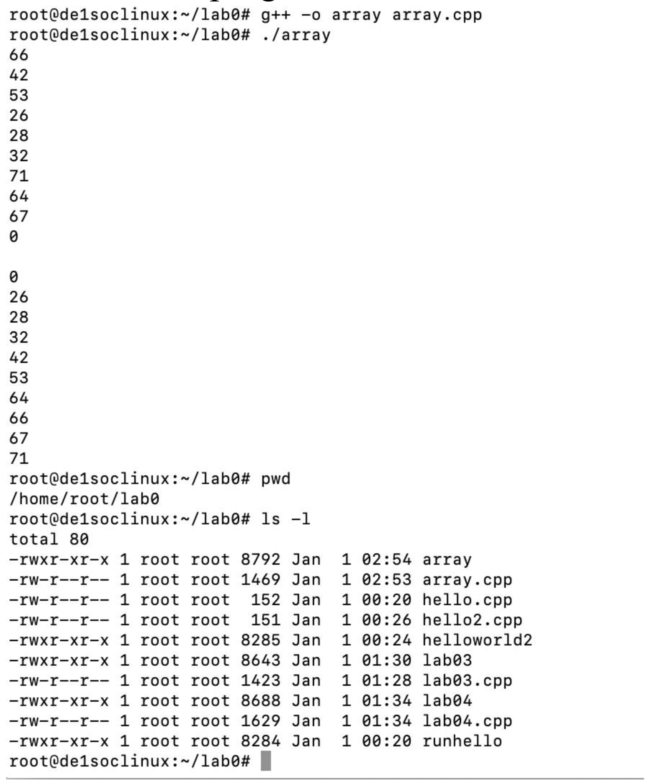
lab 0.5



```cpp
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

using namespace std;

void swap(int *min, int *current)
{
    int temp = *min;
    *min = *current;
    *current = temp;

}

void RandomArray(int v[], int size)
{
  srand(time(NULL));
  for(int i = 0; i < size; i++)
  {
      v[i] = rand() % 100;
  }
}

void PrintArray(int v[], int size)
{
  for(int i = 0; i < size; i++)
  {
  cout << v[i] << endl;
  }
}

void SortArray(int v[], int size)
{
  int min;
  // go through each element
  for(int i = 0; i < size; i++)
      {
          // set the min equal to the current element
          min = i;
          // see if it's actually the min, if not set min equal to smaller number
          for(int j = 0; j < size; j++)
          {
              if(v[min] < v[j])
              {
                  min = j;
              }
              // swap the two numbers in the array
              swap(&v[min], &v[i]);
          }
      }
}

int main()
{
    //Create an array to hold 10 integer values: randomarray
    int arraysize = 10;
    int randomarray[arraysize];
    // Generate 10 ranodm integer values and save them to randomarray
    RandomArray(randomarray, arraysize);
    // Display randomarray
    PrintArray(randomarray, arraysize);
    // Sort randomarray in ascending order
    SortArray(randomarray, arraysize);
    // Display randomarray
    cout << endl;
    PrintArray(randomarray, arraysize);
    return 0;
}
```

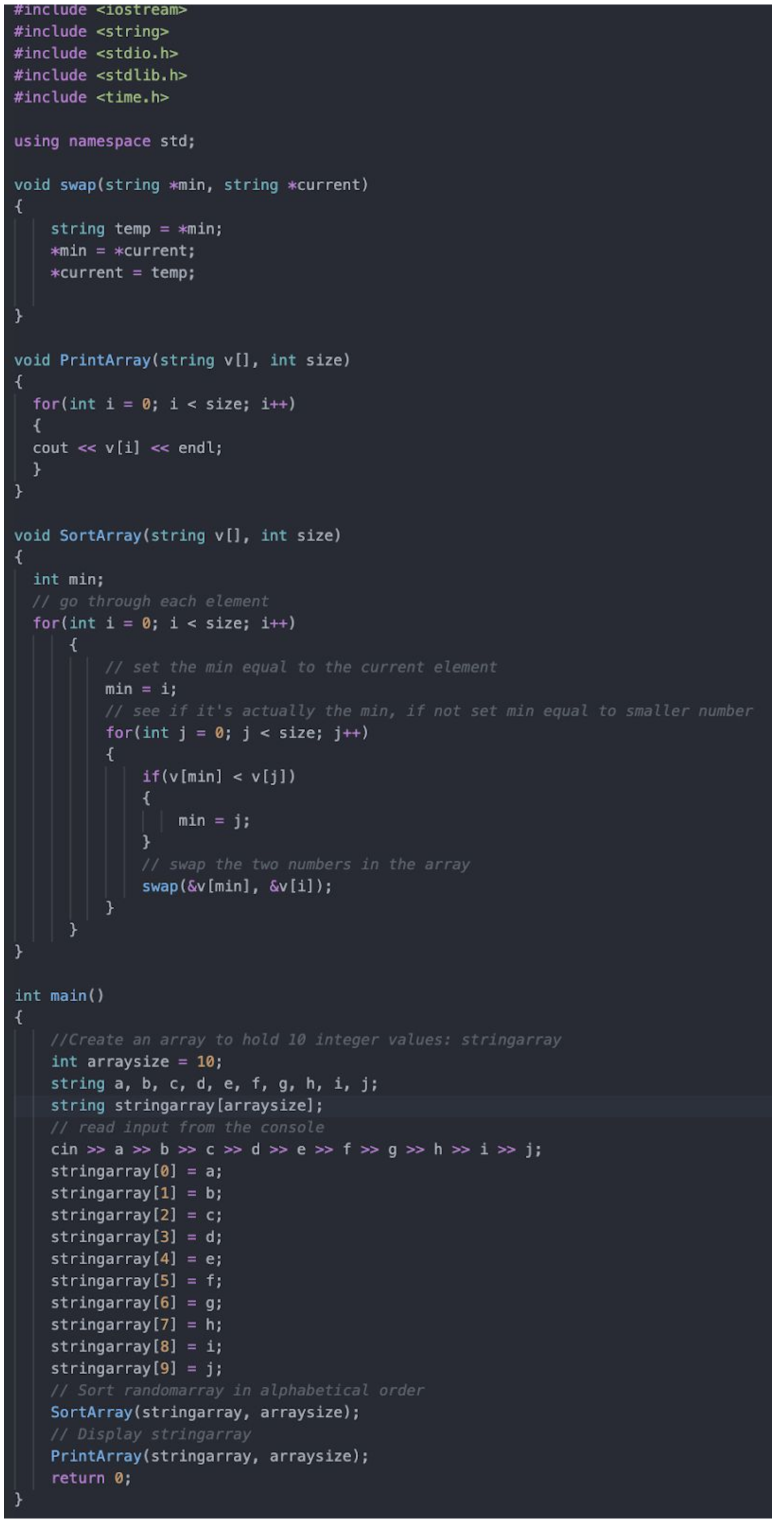
Above is the code for this section of the lab, screenshotted from the text editor. The code has been abstracted into a number of methods called from the main method of the class.

Below is the output of the method. The output is very similar to the output of lab03.cpp, except the numbers are different because they are randomly generated each time the program is run.



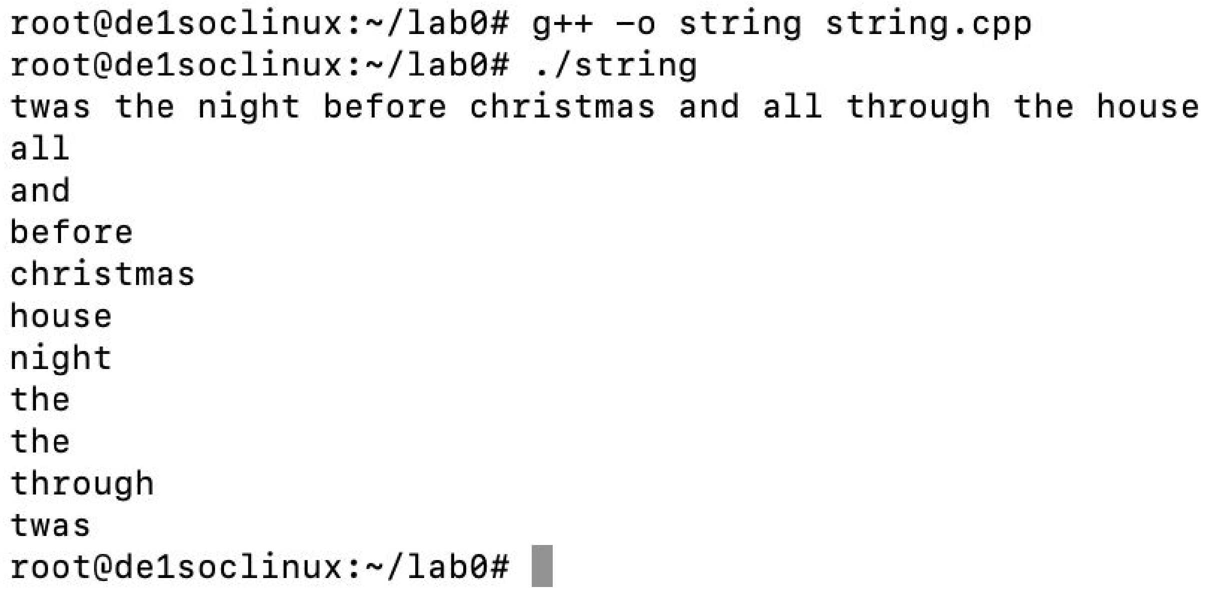
```
root@de1soclinux:~/lab0# g++ -o array array.cpp
root@de1soclinux:~/lab0# ./array
66
42
53
26
28
32
71
64
67
0

0
26
28
32
42
53
64
66
67
71
root@de1soclinux:~/lab0# pwd
/home/root/lab0
root@de1soclinux:~/lab0# ls -l
total 80
-rwxr-xr-x 1 root root 8792 Jan  1 02:54 array
-rw-r--r-- 1 root root 1469 Jan  1 02:53 array.cpp
-rw-r--r-- 1 root root  152 Jan  1 00:20 hello.cpp
-rw-r--r-- 1 root root  151 Jan  1 00:26 hello2.cpp
-rwxr-xr-x 1 root root 8285 Jan  1 00:24 helloworld2
-rwxr-xr-x 1 root root 8643 Jan  1 01:30 lab03
-rw-r--r-- 1 root root 1423 Jan  1 01:28 lab03.cpp
-rwxr-xr-x 1 root root 8688 Jan  1 01:34 lab04
-rw-r--r-- 1 root root 1629 Jan  1 01:34 lab04.cpp
-rwxr-xr-x 1 root root 8284 Jan  1 00:20 runhello
root@de1soclinux:~/lab0# 
```

Lab 0.6



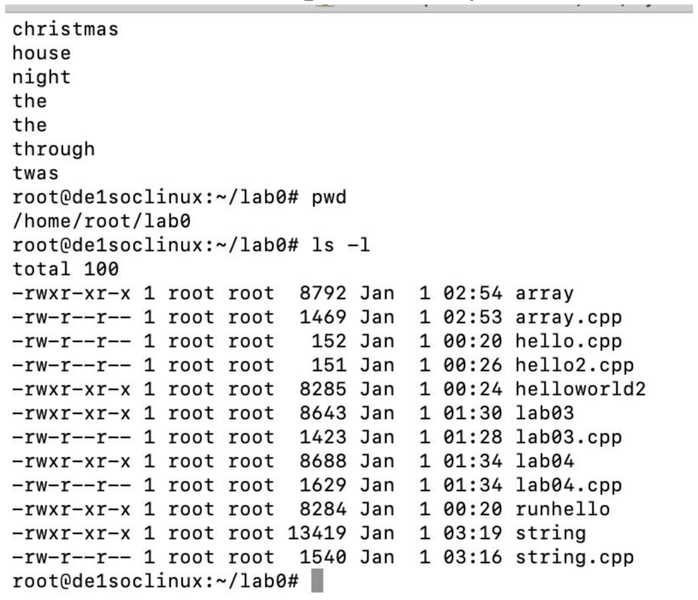
```
#include <iostream>
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

using namespace std;

void swap(string *min, string *current)
{
    string temp = *min;
    *min = *current;
    *current = temp;

}

void PrintArray(string v[], int size)
{
  for(int i = 0; i < size; i++)
  {
  cout << v[i] << endl;
  }
}

void SortArray(string v[], int size)
{
  int min;
  // go through each element
  for(int i = 0; i < size; i++)
      {
          // set the min equal to the current element
          min = i;
          // see if it's actually the min, if not set min equal to smaller number
          for(int j = 0; j < size; j++)
          {
              if(v[min] < v[j])
              {
                  min = j;
              }
              // swap the two numbers in the array
              swap(&v[min], &v[i]);
          }
      }
}

int main()
{
    //Create an array to hold 10 integer values: stringarray
    int arraysize = 10;
    string a, b, c, d, e, f, g, h, i, j;
    string stringarray[arraysize];
    // read input from the console
    cin >> a >> b >> c >> d >> e >> f >> g >> h >> i >> j;
    stringarray[0] = a;
    stringarray[1] = b;
    stringarray[2] = c;
    stringarray[3] = d;
    stringarray[4] = e;
    stringarray[5] = f;
    stringarray[6] = g;
    stringarray[7] = h;
    stringarray[8] = i;
    stringarray[9] = j;
    // Sort randomarray in alphabetical order
    SortArray(stringarray, arraysize);
    // Display stringarray
    PrintArray(stringarray, arraysize);
    return 0;
}
```

Above is the code for this portion of the lab from the text editor. The code has been adapted to instead take in 10 strings and order those alphabetically. The helper methods for this main function are unchanged except for the input types.



```
root@de1soclinux:~/lab0# g++ -o string string.cpp
root@de1soclinux:~/lab0# ./string
twas the night before christmas and all through the house
all
and
before
christmas
house
night
the
the
through
twas
root@de1soclinux:~/lab0# 
```

This above image shows the results of running this method on the set of strings “twas the night before christmas and all through the house”. Each string entered is ordered alphabetically.



```
christmas
house
night
the
the
through
twas
root@de1soclinux:~/lab0# pwd
/home/root/lab0
root@de1soclinux:~/lab0# ls -l
total 100
-rwxr-xr-x 1 root root  8792 Jan  1 02:54 array
-rw-r--r-- 1 root root  1469 Jan  1 02:53 array.cpp
-rw-r--r-- 1 root root   152 Jan  1 00:20 hello.cpp
-rw-r--r-- 1 root root   151 Jan  1 00:26 hello2.cpp
-rwxr-xr-x 1 root root  8285 Jan  1 00:24 helloworld2
-rwxr-xr-x 1 root root  8643 Jan  1 01:30 lab03
-rw-r--r-- 1 root root  1423 Jan  1 01:28 lab03.cpp
-rwxr-xr-x 1 root root  8688 Jan  1 01:34 lab04
-rw-r--r-- 1 root root  1629 Jan  1 01:34 lab04.cpp
-rwxr-xr-x 1 root root  8284 Jan  1 00:20 runhello
-rwxr-xr-x 1 root root 13419 Jan  1 03:19 string
-rw-r--r-- 1 root root  1540 Jan  1 03:16 string.cpp
root@de1soclinux:~/lab0# 
```

This image shows the path to this file and the other files within the path.

Analysis

lab 0.0

In this lab, we had a lot of difficulty setting up the connection. There was conflicting information and we had a lack of experience in setting up the board. Luckily, we were able to get some help and figured out how to properly transfer files and ssh into the board. We discovered that the workflow of coding on our mac, checking that the code compiles in an online code editor, and then transferring the files over to be run, is the most efficient technique.

lab 0.1

Lab 0.1 went smoothly and we were able to quickly get the hang of coding in C++ when making simple programs. While the output commands were confusing at first, they were easy to adapt to.

lab 0.2

Similar to lab 0.1, we were able to quickly code and compile this portion. We found it interesting that the 'std::' was optional. Additionally, removing *using namespace std;* breaking the program was an interesting result. We predicted this result but it was still interesting to see what is required for the program to function and what isn't.

lab 0.3

This portion of the lab was much more difficult than the previous two and we had to do more research into the way C++ works. One particularly tricky portion of the lab was trying to implement the selection sort algorithm, as there we were unable to get the swap() method to work. Instead, we implemented our own swap method, taking advantage of using references to change the order of the array.

lab 0.4

The actual changes necessary for this portion of the lab were few, but nonetheless it was very informative to know how to output the addresses of the various variables used in the program, and where are changing these addresses in the program.

lab 0.5

We enjoyed abstracting the various functionalities of this method into separate methods, and calling them in the main method. The actual coding was fairly simple, and it made the code significantly cleaner and more readable.

lab 0.6

It was interesting to us to see how many changes are needed in the code now that the functionality is abstracted into multiple methods. We forgot to change the signature of the methods to now take in Strings instead of Ints, but this was an easy fix. We also had some trouble with adding all of the input variables into the stringarray at the same time, and instead had to add them one by one. We still aren't sure if it's possible to add them all together.

Conclusion

The above results show that we have learned a lot about the functions of our board, such as how to connect to and share files with it, and have learned the basics of C++ and other important command line tools such as ssh, scp, and nano. Taken together, the experiences outlined in this lab report indicate that we learned a lot about the C++ language and the expectations and course goals through this lab.

Appendix

hello.cpp

```
// Example Program: Hello World
#include <iostream>
#include <string>

using namespace std;

int main()
{
std::cout << "Hello, World!\n";
return 0;
}
```

hello2.cpp

```
// Example Program: Hello World part 2
#include<iostream>
#include<string>
using namespace std;

int main()
{
cout << "Hello, World!\n";
return 0;
}
```

lab03.cpp

```
#include <iostream>
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

using namespace std;

void swap(int *min, int *current)
{
   int temp = *min;
   *min = *current;
   *current = temp;

}
```

```
int main()
{
   //Create an array to hold 10 integer values: randomarray
   int arraysize = 10;
   int randomarray[arraysize];
   // Generate 10 ranodm integer values and save them to randomarray
   srand(time(NULL));
   for(int i = 0; i < arraysize; i++)
   {
      randomarray[i] = rand() % 100;
   }
   // Display randomarray
      for(int i = 0; i < arraysize; i++)
   {
      cout << randomarray[i] << endl;
   }
   // Sort randomarray in ascending order
   int min;
   // go through each element
   for(int i = 0; i < arraysize; i++)
      {
         // set the min equal to the current element
         min = i;
         // see if it's actually the min, if not set min equal to smaller number
         for(int j = 0; j < arraysize; j++)
         {
            if(randomarray[min] < randomarray[j])
            {
               min = j;
            }
            // swap the two numbers in the array
            swap(&randomarray[min], &randomarray[i]);
         }
      }
   // Display randomarray
   cout << endl;
     for(int i = 0; i < arraysize; i++)
   {
      cout << randomarray[i] << endl;
   }
   return 0;
}
```

lab04.cpp

```
#include <iostream>
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

using namespace std;

void swap(int *min, int *current)
{
  int temp = *min;
  *min = *current;
  *current = temp;

}

int main()
{
  //Create an array to hold 10 integer values: randomarray
  int arraysize = 10;
  int randomarray[arraysize];

  // Generate 10 ranodm integer values and save them to randomarray
  srand(time(NULL));
  for(int i = 0; i < arraysize; i++)
  {
    randomarray[i] = rand() % 100;
  }
  // Display randomarray
    for(int i = 0; i < arraysize; i++)
  {
    cout << randomarray[i] << endl;
  }
  // Sort randomarray in ascending order
  int min;
  // go through each element
  for(int i = 0; i < arraysize; i++)
    {
      // set the min equal to the current element
      min = i;
      // see if it's actually the min, if not set min equal to smaller number
```

```
        for(int j = 0; j < arraysize; j++)
        {
          if(randomarray[min] < randomarray[j])
          {
            min = j;
          }
          // swap the two numbers in the array
          swap(&randomarray[min], &randomarray[i]);
        }
      }
   // Display randomarray
   cout << endl;
     for(int i = 0; i < arraysize; i++)
   {
      cout << randomarray[i] << endl;
   }

   // display addresses
   cout << &arraysize << endl;
   cout << &randomarray << endl;
   cout << &min << endl;
   for(int i = 0; i < arraysize; i++)
   {
    cout << &randomarray[i] << endl;
   }

  return 0;
}
```

array.cpp

```
#include <iostream>
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

using namespace std;

void swap(int *min, int *current)
{
   int temp = *min;
   *min = *current;
   *current = temp;
```

```
}

void RandomArray(int v[], int size)
{
 srand(time(NULL));
 for(int i = 0; i < size; i++)
 {
    v[i] = rand() % 100;
 }
}

void PrintArray(int v[], int size)
{
 for(int i = 0; i < size; i++)
 {
 cout << v[i] << endl;
 }
}

void SortArray(int v[], int size)
{
 int min;
 // go through each element
 for(int i = 0; i < size; i++)
    {
       // set the min equal to the current element
       min = i;
       // see if it's actually the min, if not set min equal to smaller number
       for(int j = 0; j < size; j++)
       {
          if(v[min] < v[j])
          {
             min = j;
          }
          // swap the two numbers in the array
          swap(&v[min], &v[i]);
       }
    }
}

int main()
{
   //Create an array to hold 10 integer values: randomarray
   int arraysize = 10;
   int randomarray[arraysize];
   // Generate 10 ranodm integer values and save them to randomarray
   RandomArray(randomarray, arraysize);
   // Display randomarray
   PrintArray(randomarray, arraysize);
   // Sort randomarray in ascending order
   SortArray(randomarray, arraysize);
   // Display randomarray
   cout << endl;
   PrintArray(randomarray, arraysize);
   return 0;
}
```

string.cpp

```
#include <iostream>
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

using namespace std;

void swap(string *min, string *current)
{
   string temp = *min;
   *min = *current;
   *current = temp;

}

void PrintArray(string v[], int size)
{
  for(int i = 0; i < size; i++)
  {
  cout << v[i] << endl;
  }
}

void SortArray(string v[], int size)
{
  int min;
```

```
 // go through each element
 for(int i = 0; i < size; i++)
    {
       // set the min equal to the current element
       min = i;
       // see if it's actually the min, if not set min equal to smaller number
       for(int j = 0; j < size; j++)
       {
          if(v[min] < v[j])
          {
             min = j;
          }
          // swap the two numbers in the array
          swap(&v[min], &v[i]);
       }
    }
}

int main()
{
   //Create an array to hold 10 integer values: stringarray
   int arraysize = 10;
   string a, b, c, d, e, f, g, h, i, j;
   string stringarray[arraysize];
   // read input from the console
   cin >> a >> b >> c >> d >> e >> f >> g >> h >> i >> j;
   stringarray[0] = a;
   stringarray[1] = b;
   stringarray[2] = c;
   stringarray[3] = d;
   stringarray[4] = e;
   stringarray[5] = f;
   stringarray[6] = g;
   stringarray[7] = h;
   stringarray[8] = i;
   stringarray[9] = j;
   // Sort randomarray in alphabetical order
   SortArray(stringarray, arraysize);
   // Display stringarray
   PrintArray(stringarray, arraysize);
   return 0;
}
```