

CSC112 Spring 2011
Fundamentals of Computer Science
Lab 5 – Binary files, 1-D Arrays

Part 0: Turning off the Update Reminder

Unfortunately, updating your Ubuntu virtual machine interferes with the Virtual Box Extensions. As a result, full screen mode no longer works after running the updates. To fix this, we will do two things:

1. Turn off the update reminder. Using the menus in the upper left, go to: System → Preferences Startup Applications. A window will pop up labeled “Startup Applications Preferences”. Scroll down to find an entry labeled “Update Notifier”. Click on the check box on the left, and the check mark should disappear. Click on close, and you are done.
2. Restore full-screen mode to working condition. Mr. Whitner will be attending lab, and will help students individually to get full-screen mode working again.

Part I: 1-D Dynamically Allocated Arrays

Create a directory named Lab5. Keep all of your source and compiled programs in the directory Lab5.

Write a C++ program that reverses a binary audio file named “audio.s16”. To do this there are several details.

1. You can download the file “audio.s16” from the URL:

<http://menehune.opt.wfu.edu/CSC112/Lab5/audio.s16>

2. Use the techniques discussed in class for reading and writing binary files using objects of type **ifstream** and **ofstream**. The file consists of a sequence of 16 bit signed integers. The base type for this data in C++ is “**short int**”.
3. Your program will read the entire file and store it in a single array. To allocate this array, you will need to know how many bytes in the file. Use the system function **stat()** to get this information. See the example handout for help using **stat()**. Your array will be an array with base type **short int**. You will need to compute the number of **short ints** that correspond to the file size: divide the number of bytes by the size of a **short int**. Use the built-in C++ function **sizeof(short int)** to get the size of type **short int**.
4. You should create a dynamically allocated array using a pointer variable, **new**, and **delete** to manage storage for the contents of the input file.
5. Your program must check for errors, including:
 - (a) Check the return code from the call to **stat()** to ensure that the call succeeded. The function **stat** will return 0 to indicate success and -1 to indicate an error.
 - (b) When opening both the input and output files, check for failure, and issue appropriate error messages in case of failure.

- (c) When reading from a file , and when reading to a file (using “.read()” and “.write()” member functions of an ifstream and an ofstream object respectively), check for failure to read the requested number of bytes, and issue an error message if appropriate. Use the “.fail” member function to check for failure.
6. Write a function reverse() to reverse the array. Your function should reverse the array “in place”. I.e., reverse the array by swapping corresponding elements. Do not allocate a secondary array. Your function header should be:

```
void reverse( short int a[], int n )
```

7. Write the reversed array to a binary file named “audio_rev.s16”.
8. Install the **sox** package on your Ubuntu virtual machine.
- (a) Use the command: `sudo apt-get install sox`
9. Use the play command to listen to both the forward and reversed audio. The simple audio format we are using does not store the sample rate in the file, so we have to supply that information on the command line of the play command. The sample rate is the standard CD audio rate of 44100 samples per second. To listen to the downloaded audio file, use the command:

```
play -r 44100 audio.s16
```

After running your program, you can listen to the sound sample backwards using:

```
play -r 44100 audio_rev.s16
```

10. It is time for us to work on programming style.
- (a) Use comments in your program to describe the logic of what your program is doing.
- (b) Indent all control constructs for easier reading.

Turn in: Change to the directory containing the sub-directory “Lab5” Create a file named “lab5.tar” using the command:

```
tar cf lab5.tar Lab5
```

Upload the file “lab5.tar” to your account on telesto.