

CS2263

Lab 2

By:

Christian Quinn

&

Logan Fitzpatrick

Git Repo Link : https://github.com/Christian862/CS2263_Summer2019_L2

Code for ArraySort.c after modification

```
/*
 *
 * ArraySort.c
 *
 * Created by Jean-Philippe Legault
 *
 * Your task is to implement in place sorting using the two available
functions
 * swapAdjacent, and compareAdjacent.
 *
 * Some bug might have been introduced... you will have to find out if
there are any!
 * if so, you will have to correct it
 *
 *****/
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

void printArray(int *array, int size)
{
    for(int i=0; i<size; i++)
    {
        if(i != 0)
        {
            printf(", ");
        }
        printf("%d", array[i]);
    }
    printf("\n");
}

void swapAdjacent(int *a, int index)
{
    int temp = *(a + index);
    *(a + index) = *(a + index + 1);
    *(a + index + 1) = temp;
}

int compareAdjacent(int *a, int index)
{
    return *(a + index) - *(a + index + 1);
}

/**
 * TODO: implement in place sorting on an array
 * by using the two functions swapAdjacent and compareAdjacent
 */
void inplaceSort(int * arr, int size)
{
    for(int i = 0; i < size; i++){
        for(int j = 0; j < size-1; j++){
```

```

        if(compareAdjacent(arr, j) > 0){
            swapAdjacent(arr, j);
        }
    }
}

int main(void)
{
    int array_size = 0;
    printf("Enter the array size (>0) and the numbers to fill the array
with: ");
    if(!scanf("%d", &array_size))
    {
        printf("ERROR. Must enter an integer.\n");
        return EXIT_FAILURE;
    }
    else if(array_size < 1)
    {
        printf("ERROR. array size must be at least 1.\n");
        return EXIT_FAILURE;
    }

    int a[array_size];

    /*****
    * TODO finish parsing the user input to fill the array
    *
    * it should parse user input with scanf to fill the array with values
    *****/
    int i, inp;
    for(i = 0; i < array_size; i++) {
        if(scanf("%d", &inp) == 1){
            a[i] = inp;
        }
        else{
            printf("Invalid input\n");
        }
    }

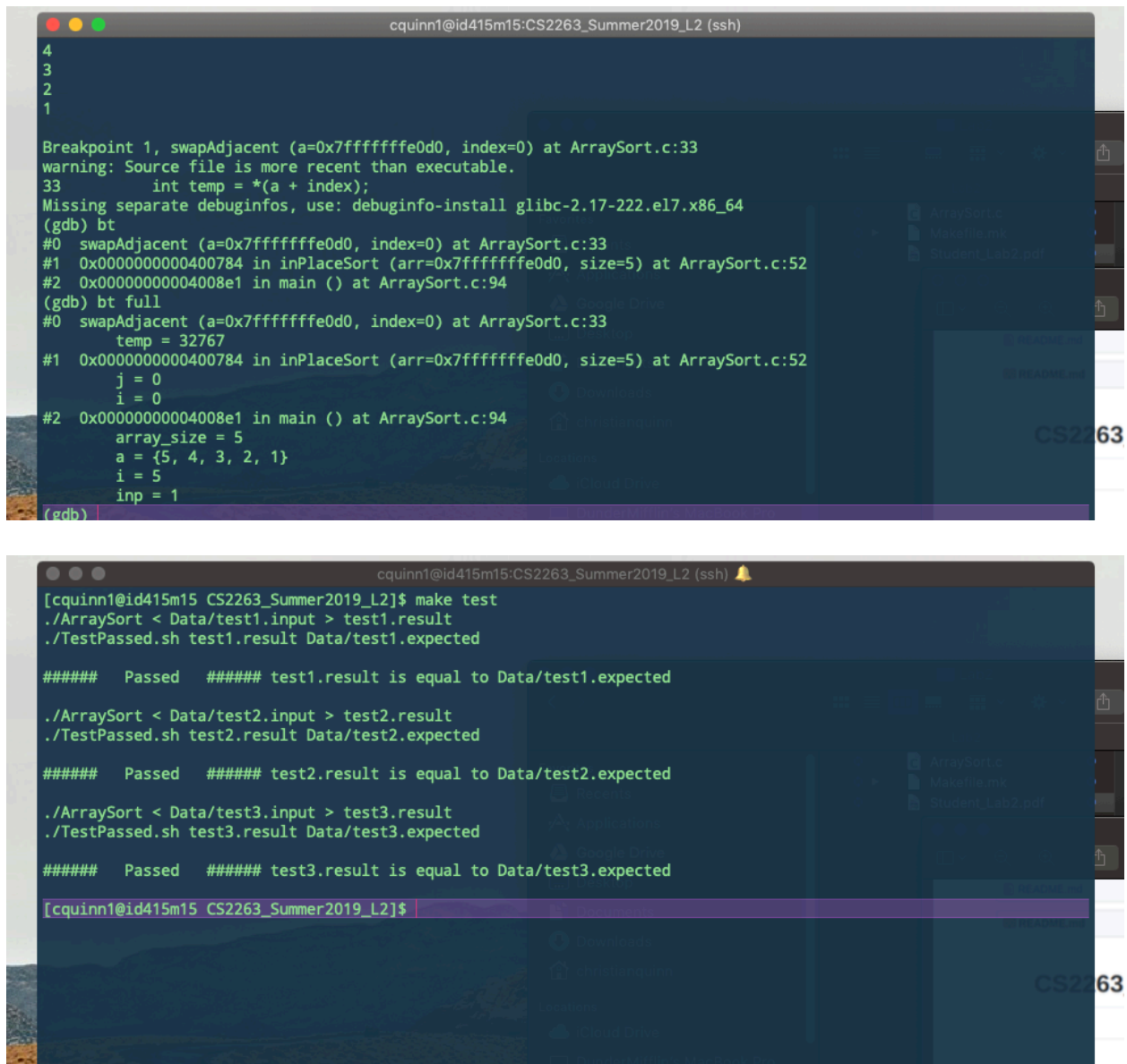
    //printf("=== Array before Sorting = ");
    printArray(a, array_size);

    inplaceSort(a, array_size);

    //printf("=== Array after Sorting = ");
    printArray(a, array_size);
}

```

Terminal Recording of tests via make file and GDB



The image consists of two terminal screenshots. The top screenshot shows a GDB session where a breakpoint is set at line 33 of ArraySort.c. The user then runs the program, and the debugger stops at the breakpoint. The user enters 'bt' to see the backtrace, which shows the call stack: swapAdjacent, inPlaceSort, and main. The user then enters 'bt full' to see more details, including the values of variables like 'temp', 'j', 'i', 'array_size', 'a', and 'inp'. The bottom screenshot shows the user running 'make test' in the terminal. The output shows three tests being run, each passing. The tests compare the output of ArraySort with expected results in Data/test1.expected, Data/test2.expected, and Data/test3.expected.

```
cquinn1@id415m15:CS2263_Summer2019_L2 (ssh)
4
3
2
1

Breakpoint 1, swapAdjacent (a=0x7fffffff0d0, index=0) at ArraySort.c:33
warning: Source file is more recent than executable.
33      int temp = *(a + index);
Missing separate debuginfos, use: debuginfo-install glibc-2.17-222.el7.x86_64
(gdb) bt
#0  swapAdjacent (a=0x7fffffff0d0, index=0) at ArraySort.c:33
#1  0x000000000400784 in inPlaceSort (arr=0x7fffffff0d0, size=5) at ArraySort.c:52
#2  0x0000000004008e1 in main () at ArraySort.c:94
(gdb) bt full
#0  swapAdjacent (a=0x7fffffff0d0, index=0) at ArraySort.c:33
    temp = 32767
#1  0x000000000400784 in inPlaceSort (arr=0x7fffffff0d0, size=5) at ArraySort.c:52
    j = 0
    i = 0
#2  0x0000000004008e1 in main () at ArraySort.c:94
    array_size = 5
    a = {5, 4, 3, 2, 1}
    i = 5
    inp = 1
(gdb)

[cquinn1@id415m15 CS2263_Summer2019_L2]$ make test
./ArraySort < Data/test1.input > test1.result
./TestPassed.sh test1.result Data/test1.expected

##### Passed ##### test1.result is equal to Data/test1.expected

./ArraySort < Data/test2.input > test2.result
./TestPassed.sh test2.result Data/test2.expected

##### Passed ##### test2.result is equal to Data/test2.expected

./ArraySort < Data/test3.input > test3.result
./TestPassed.sh test3.result Data/test3.expected

##### Passed ##### test3.result is equal to Data/test3.expected

[cquinn1@id415m15 CS2263_Summer2019_L2]$
```