Lab 5

Exercise One

Make command output for a successful compile

```
[mmoustaf@gc112m38 Lab5]$ make Strings.o
gcc -w -g -Wall -Wshadow -c Strings.c
```

Exercise Two

stringTest1.c

```c
#include "Strings.h"
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[])
{
    String copy = duplicateString(argv[0]);
    printf("%s\n", copy);
    return EXIT_SUCCESS;
}
```

The output from running the Makefile

```
[mmoustaf@gc112m38 Lab5]$ make valgrindsT1
gcc -w -g -Wall -Wshadow -c Strings.c
gcc -w -g -Wall -Wshadow -o stringTest1 stringTest1.o Strings.o
valgrind --tool=memcheck --leak-check=full --verbose --log-file=log.txt ./stringTest1
./stringTest1
```

Log.txt before free

```
==16534== HEAP SUMMARY:
==16534==     in use at exit: 9 bytes in 1 blocks
==16534==   total heap usage: 1 allocs, 0 frees, 9 bytes allocated
==16534==
==16534== Searching for pointers to 1 not-freed blocks
==16534== Checked 70,304 bytes
==16534==
==16534== 9 bytes in 1 blocks are definitely lost in loss record 1 of 1
==16534==    at 0x4C29F73: malloc (vg_replace_malloc.c:309)
==16534==    by 0x40070C: mallocString (Strings.c:6)
==16534==    by 0x400754: duplicateString (Strings.c:21)
==16534==    by 0x4006DA: main (stringTest1.c:6)
==16534==
==16534== LEAK SUMMARY:
==16534==    definitely lost: 9 bytes in 1 blocks
==16534==    indirectly lost: 0 bytes in 0 blocks
==16534==      possibly lost: 0 bytes in 0 blocks
==16534==    still reachable: 0 bytes in 0 blocks
==16534==         suppressed: 0 bytes in 0 blocks
```

Logf.txt (After free)

```
==16945== HEAP SUMMARY:
==16945==     in use at exit: 0 bytes in 0 blocks
==16945==   total heap usage: 1 allocs, 1 frees, 9 bytes allocated
==16945==
==16945== All heap blocks were freed -- no leaks are possible
==16945==
==16945== ERROR SUMMARY: 14 errors from 4 contexts (suppressed: 0 from 0)
```

stringTest1.c

```c
#include "Strings.h"
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[])
{
    String copy = duplicateString(argv[0]);
    printf("%s\n", copy);
    freeString(copy);
    return EXIT_SUCCESS;
}
```

A screen-shot of the program running (with freeString()) and its output

```
[mmoustaf@gc112m38 Lab5]$ ./stringTest1
./stringTest1
```

A screen-shot of the program running (without freeString()) and its output

```
[mmoustaf@gc112m38 Lab5]$ gcc -o stringTest1 stringTest1.o Strings.o
[mmoustaf@gc112m38 Lab5]$ ./stringTest1
./stringTest1
```

Exercise 3

the output from running your Makefile

```
[mmoustaf@gc112m38 Lab5]$ make valgrindsLT
gcc -w -g -Wall -Wshadow -c stringListTest.c Strings.c
gcc -w -g -Wall -Wshadow -o stringListTest stringListTest.o Strings.o
valgrind --tool=memcheck --leak-check=full --verbose --log-file=logSLT.txt ./stringListTest
./stringListTest
```

the log-file outputs from the final run of valgrind

```
==19660== HEAP SUMMARY:
==19660==     in use at exit: 0 bytes in 0 blocks
==19660==   total heap usage: 1 allocs, 1 frees, 8 bytes allocated
==19660==
==19660== All heap blocks were freed -- no leaks are possible
==19660==
==19660== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

stringListTest.c

```c
#include "Strings.h"
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[])
{
    String* list = duplicateStringList(argc, argv);
    int i;
    for (i =0; i < argc; i++)
    {
        printf("%s\n", *(list+i));
        // freeString(*(list+i));
    }
    free(list);
    return EXIT_SUCCESS;

}
```

A screenshot of the program running and its output

```
[mmoustaf@gc112m38 Lab5]$ gcc -o stringListTest stringListTest.c Strings.c
[mmoustaf@gc112m38 Lab5]$ ./stringListTest dafsjksdfj
./stringListTest
dafsjksdfj
```

Exercise 4

the output from running your Makefile

```
[mmoustaf@gc112m38 Lab5]$ make valgrindsLST
gcc -w -g -Wall -Wshadow -c stringListSortTest.c Strings.c
gcc -w -g -Wall -Wshadow -o stringListSortTest stringListSortTest.o Strings.o
valgrind --tool=memcheck --leak-check=full --verbose --log-file=logSLST.txt ./stringListSortTest
./stringListSortTest
```

the log-file outputs from the final run of valgrind

```
==19974== HEAP SUMMARY:
==19974==     in use at exit: 0 bytes in 0 blocks
==19974==   total heap usage: 1 allocs, 1 frees, 8 bytes allocated
==19974==
==19974== All heap blocks were freed -- no leaks are possible
==19974==
==19974== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

stringListSortTest.c

```c
#include "Strings.h"
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
int main(int argc, char* argv[])
{
    String* list = duplicateStringList(argc, argv);
    int i;
    qsort(list,argc,sizeof(String),compareStrings);
    for (i =0; i < argc; i++)
    {
        printf("%s\n", *(list+i));
        // freeString(*(list+i));
    }
    free(list);
    return EXIT_SUCCESS;

}
```

A screen shot of the program running and its output

```
[mmoustaf@gc112m38 Lab5]$ ./stringListSortTest S560 Rolls Royse Bently
./stringListSortTest
Bently
Rolls
Royse
S560
```

Exercise 5

the screenshot of you pushing the program source to the FCS git

```
[mmoustaf@gc112m38 cs2263-mmoustaf]$ git add Strings.c Strings.h
[mmoustaf@gc112m38 cs2263-mmoustaf]$ git commit -m "Adding Strings Module"
# On branch master
nothing to commit, working directory clean
[mmoustaf@gc112m38 cs2263-mmoustaf]$ ls
arithmetic1.c  fgsmain.c  main.c  Strings.c  Strings.h  Three.c  Two.c  wrongindex.c
[mmoustaf@gc112m38 cs2263-mmoustaf]$ git commit -m "Adding Strings Module"
# On branch master
nothing to commit, working directory clean
[mmoustaf@gc112m38 cs2263-mmoustaf]$ git push origin master
Username for 'https://vcs.cs.unb.ca': mmoustaf
Password for 'https://mmoustaf@vcs.cs.unb.ca':
Everything up-to-date
[mmoustaf@gc112m38 cs2263-mmoustaf]$
```

the modified source code for the Strings module, including the Makefile

Strings.h

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

#ifndef STRINGS_H
#define STRINGS_H
typedef char *String;
// a cover function for malloc()
// malloc and return memory for a string of stringsize characters
// return (char*)NULL on failure
String mallocString(int stringsize);

// just a cover function for free()
void freeString(String s);

// create a duplicate string of s
// return it
// return (char*)NULL on failure
// should call mallocString(), and then strcpy()
String duplicateString(String s);
String *duplicateStringList(int i, String *sl);
int compareStrings(void *s1, void *s2);
String getString();
#endif
```

Strings.c

```c
#include "Strings.h"
#include <stdio.h>
#include <stdlib.h>
String mallocString(int stringsize)
{
    String pc = (String)malloc(sizeof(char) * (stringsize + 1));
    if (pc == (String)NULL)
    {
        return (String)NULL;
    }
    return pc;
}

void freeString(String s)
{
    free(s);
}

String duplicateString(String s)
{
    String copy = mallocString(sizeof(s));
    if (copy == (String)NULL)
    {
        return (String)NULL;
    }
    strcpy(copy, s);
    return copy;
}
String *duplicateStringList(int i, String *sl)
{
    String *copy = (String *)malloc(sizeof(String) * i);
    int j;
    for (j = 0; j < i; j++)
    {
        copy[j] = sl[j];
    }
    return copy;
```

```
37      return copy;
38  }
39  int compareStrings(void *s1, void *s2)
40  {
41      String *sc1 = (String*)s1;
42      String *sc2 = (String*)s2;
43      return strcmp(*sc1, *sc2);
44  }
45
46  String getString()
47  {
48      String s;
49      scanf("%[^\n]", s);
50      return s;
51  }
52
```

Makefile

```
1  Strings.o: Strings.c Strings.h
2      gcc -w -g -Wall -Wshadow -c Strings.c
3  stringTest1.o: stringTest1.c Strings.c
4      gcc -w -g -Wall -Wshadow -c stringTest1.c Strings.c
5  stringTest1: stringTest1.o Strings.o
6      gcc -w -g -Wall -Wshadow -o stringTest1 stringTest1.o Strings.o
7  run: stringTest1
8      ./stringTest1
9  valgrindsT1: stringTest1
10     valgrind --tool=memcheck --leak-check=full --verbose --log-file=logf.txt ./stringTest1
11 stringListTest.o: stringListTest.c Strings.c
12     gcc -w -g -Wall -Wshadow -c stringListTest.c Strings.c
13 stringListTest: stringListTest.o Strings.o
14     gcc -w -g -Wall -Wshadow -o stringListTest stringListTest.o Strings.o
15 valgrindsLT: stringListTest
16     valgrind --tool=memcheck --leak-check=full --verbose --log-file=logSLT.txt ./stringListTest
17 stringListSortTest.o: stringListSortTest.c Strings.c
18     gcc -w -g -Wall -Wshadow -c stringListSortTest.c Strings.c
19 stringListSortTest: stringListSortTest.o Strings.o
20     gcc -w -g -Wall -Wshadow -o stringListSortTest stringListSortTest.o Strings.o
21 valgrindsLST: stringListSortTest
22     valgrind --tool=memcheck --leak-check=full --verbose --log-file=logSLST.txt ./stringListSortTest
```