CS2263 Lab 4
Stephen Cole
3553803

makefile

```
COMPILER = gcc
C_FLAGS = -Wall -Wextra


stest: Strings.o stringTest.o
	$(COMPILER) $(C_FLAGS) -o stest stringTest.o Strings.o

Strings.o: Strings.c
	$(COMPILER) $(C_FLAGS) -c Strings.c

stringTest.o: stringTest.c
	$(COMPILER) $(C_FLAGS) -c stringTest.c
```

Strings.c

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

#define MAX_STRING_LENGTH 100

char* mallocString(int stringsize){
   char *strMem = (char *)malloc(sizeof(char) * stringsize);

   if (strMem == (char *)NULL)
   {
      fprintf(stderr, "Memory allocation failed. Program terminating.");
      return (char*)NULL;
   }

   return strMem;
}

void freeString(char *s){
   free(s);
}

// create a duplicate string of s
// return it
```

```
// return (char*)NULL on failure
// should call mallocString(), and then strcpy()
char *duplicateString(char *s){
   int stringSize = 0;

   while (s[stringSize] != '\0')
      stringSize++;

   char *dupeString = mallocString(stringSize);
   strcpy(dupeString, s);
   if(dupeString == (char *)NULL)
   {
      fprintf(stderr, "Memory allocation failed. Program terminating.");
      return (char *)NULL;
   }

   return dupeString;
}

int fputString(FILE* pFOut, char* s)
{
        int i;

        for(i=0; i<MAX_STRING_LENGTH; i++)
        {
                if(s[i] == '\n')
                {
                        fprintf(pFOut, "%d\n", i);
                break;
                }

                fprintf(pFOut, "%c", s[i]);
        }
        return i;
}

char* fgetString(FILE* pFIn)
{
        char* str;
        str = mallocString(MAX_STRING_LENGTH);

        if( fgets(str, 100, pFIn) == NULL)
                return NULL;

        char* dupStr;
        dupStr = duplicateString(str);
```

```
	free(str);

	return dupStr;
}
```

Strings.h

```
#ifndef STRINGS_H
#define STRINGS_H

// a cover function for malloc()
// malloc and return memory for a string of stringsize characters
// return (char*)NULL on failure
char* mallocString(int stringsize);

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

// create a duplicate string of s
// return it
// return (char*)NULL on failure
// should call mallocString(), and then strcpy()
char* duplicateString(char* s);

int fputString(FILE* pFOut, char* s);

char* fgetString(FILE* pFIn);

#endif
```

stringTest.c

```
#include <stdio.h>
#include <stdlib.h>
#include "Strings.h"

#define MAX_STRING_LENGTH 100

int main(int argc, char* argv[]){

	char* string1;
	char* string2;
	FILE* fp = fopen(argv[1], "w");
	string1 = mallocString(MAX_STRING_LENGTH);
	string2 = mallocString(MAX_STRING_LENGTH);
	int length1;
```

```
    int length2;

    if( fgets(string1, 100, stdin) != NULL )
    {
    length1 = fputString(fp, string1);
    }

    if( fgets(string2, 100, stdin) != NULL )
{
    length2 = fputString(fp, string2);
}

    printf("length1: %d, length2: %d\n", length1, length2);


    free(string2);
    free(string1);
    fclose(fp);

    char* str1;
    char* str2;
    FILE* fpr = fopen(argv[1], "r");

    str1 = fgetString(fpr);
    str2 = fgetString(fpr);
    printf("String1: %sString2: %s", str1, str2);

    return EXIT_SUCCESS;
}
```

in.txt

first line
second line

Q0

```
[scole4@gaea lab4]$ make
gcc -Wall -Wextra -c Strings.c
gcc -Wall -Wextra -c stringTest.c
stringTest.c: In function ‘main’:
stringTest.c:5:14: warning: unused parameter ‘argc’ [-Wunused-param
 int main(int argc, char* argv[]){
              ^
gcc -Wall -Wextra -o stest stringTest.o Strings.o
[scole4@gaea lab4]$ ./stest
./stest
[scole4@gaea lab4]$
```

Q1

```
[scole4@gaea lab4]$ make
gcc -Wall -Wextra -c Strings.c
gcc -Wall -Wextra -c stringTest.c
stringTest.c: In function ‘main’:
stringTest.c:7:14: warning: unused parameter ‘argc’ [-Wunused-param
 int main(int argc, char* argv[]){
              ^
gcc -Wall -Wextra -o stest stringTest.o Strings.o
```

```
[scole4@gaea lab4]$ ./stest out < in.txt
length1: 10, length2: 11
```

out.txt

first line10
second line11

Q2

```
[scole4@gaea lab4]$ make
gcc -Wall -Wextra -c Strings.c
gcc -Wall -Wextra -c stringTest.c
stringTest.c: In function ‘main’:
stringTest.c:7:14: warning: unused parameter ‘argc’ [-Wunused-parame
 int main(int argc, char* argv[]){
              ^
gcc -Wall -Wextra -o stest stringTest.o Strings.o
```

```
[scole4@gaea lab4]$ ./stest out < in.txt
length1: 10, length2: 11
String1: first line10
String2: second line11
```

Q4

```
[scole4@gaea cs2263-scole4]$ git commit -m "lab4"
[master dff2485] lab4
 15 files changed, 160 insertions(+)
 rename arithmetic1.c => lab3/arithmetic1.c (100%)
 rename arrindex.c => lab3/arrindex.c (100%)
 rename loopbyaddress.c => lab3/loopbyaddress.c (100%)
 rename test => lab3/test (100%)
 rename wrongindex.c => lab3/wrongindex.c (100%)
 rename wrongindexoriginal.c => lab3/wrongindexoriginal.c (100%)
 create mode 100644 lab4/Strings.c
 create mode 100644 lab4/Strings.h
 create mode 100644 lab4/Strings.o
 create mode 100644 lab4/in.txt
 create mode 100644 lab4/makefile
 create mode 100644 lab4/out
 create mode 100755 lab4/stest
 create mode 100644 lab4/stringTest.c
 create mode 100644 lab4/stringTest.o
[scole4@gaea cs2263-scole4]$ git push origin master
Username for 'https://vcs.cs.unb.ca': scole4
Password for 'https://scole4@vcs.cs.unb.ca':
Counting objects: 13, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 6.79 KiB | 0 bytes/s, done.
Total 12 (delta 0), reused 0 (delta 0)
To https://vcs.cs.unb.ca/git/cs2263-scole4
   33c8485..dff2485  master -> master
```