

# Lab 4

## Exercise 1

strings.c source code

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "Strings.h"
4  #include <string.h>
5
6  char *mallocString(int stringsize)
7  {
8      char *pc = (char *)malloc(sizeof(char) * (stringsize + 1));
9      if (pc == (char *)NULL)
10     {
11         return (char *)NULL;
12     }
13     return pc;
14 }
15
16 void freeString(char *s)
17 {
18     free(s);
19 }
20
21 char *duplicateString(char *s)
22 {
23     char *copy = mallocString(sizeof(s));
24     if (copy == (char *)NULL)
25     {
26         return (char *)NULL;
27     }
```

```
27     }
28     strcpy(copy, s);
29     return copy;
30 }
31 int fputString(FILE *pFOut, char *s)
32 {
33     fprintf(pFOut, "%d %s", strlen(s)-1, s);
34 }
35
```

main.c Source code

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "Strings.h"
4  #include <string.h>
5  int main(int argc, char *argv[])
6  {
7      FILE *fI = fopen("input.txt", "r");
8      if (fI == (FILE*)NULL)
9      {
10         printf("Error opening input.txt\n");
11     }
12
13     FILE *fO = fopen(argv[1], "w");
14     if (fO == (FILE*)NULL)
15     {
16         printf("Error opening %s\n", argv[1]);
17     }
18     char* s = fgets(stdin, 100, fI);
19     fputString(fO, s);
20     if(*(s+strlen(s)-1) == '\n')
21     {
22         s = fgets(stdin, 100, fI);
23         fputString(fO, s);
24     }
25     fclose(fI);
26     fclose(fO);
27     return EXIT_SUCCESS;
28 }
29
```

## Makefile for Exercise 1

```
1 Strings.o: Strings.c Strings.h
2 gcc -c Strings.c
3 main: main.o Strings.o
4 gcc -o main main.o Strings.o
5 main.o: main.c
6 gcc -c main.c
7 run: main
8 ./main output.txt
9
```

## Screenshot of make command at work

```
[mmoustaf@gc112m38 Lab4]$ make
make: `Strings.o' is up to date.
[mmoustaf@gc112m38 Lab4]$
```

```
[mmoustaf@gc112m38 Lab4]$ make run
gcc -c main.c
main.c: In function 'main':
main.c:18:5: warning: passing argument 1 of 'fgets' from incompatible pointer type [enabled by default]
    char* s = fgets(stdin, 100, fI);
    ^
In file included from main.c:1:0:
/usr/include/stdio.h:622:14: note: expected 'char * __restrict__' but argument is of type 'struct _IO_FILE *'
extern char *fgets (char * __restrict __s, int __n, FILE * __restrict __stream)
                  ^
main.c:22:7: warning: passing argument 1 of 'fgets' from incompatible pointer type [enabled by default]
    s = fgets(stdin, 100, fI);
    ^
In file included from main.c:1:0:
/usr/include/stdio.h:622:14: note: expected 'char * __restrict__' but argument is of type 'struct _IO_FILE *'
extern char *fgets (char * __restrict __s, int __n, FILE * __restrict __stream)
                  ^
gcc -o main main.o Strings.o
./main output.txt
[mmoustaf@gc112m38 Lab4]$
```

## Input file

```
I love C!
Hello!
```

Output file

```
9 I love C!  
6 Hello!
```

---

Exercise 2

## Strings.c source code

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "Strings.h"
4  #include <string.h>
5
6  char *mallocString(int stringsize)
7  {
8      char *pc = (char *)malloc(sizeof(char) * (stringsize + 1));
9      if (pc == (char *)NULL)
10     {
11         return (char *)NULL;
12     }
13     return pc;
14 }
15
16 void freeString(char *s)
17 {
18     free(s);
19 }
20
21 char *duplicateString(char *s)
22 {
23     char *copy = mallocString(sizeof(s));
24     if (copy == (char *)NULL)
25     {
26         return (char *)NULL;
27     }
28     strcpy(copy, s);
29     return copy;
```

```

29     return copy;
30 }
31 int fputString(FILE *pFOut, char *s)
32 {
33     fprintf(pFOut, "%d %s",strlen(s)-1, s);
34 }
35 char* fgetString(FILE* pFIn)
36 {
37     int length;
38     fscanf(pFIn, "%d", &length);
39     char *stringp = (char*) mallocString(sizeof(char) * length+2);
40
41     fgets(stringp, length+2, pFIn); //for NULL
42     int i;
43     for (i = 1; i < length+2; i++){
44         printf("%c", stringp[i]);
45     }
46     printf("\n");
47
48     while(stringp[i] != EOF){
49         while (i <= length+2 )
50         {
51             fscanf(pFIn, "%d", &length);
52             stringp = (char*) mallocString(sizeof(char) * length+2);
53             fgets(stringp, length+3, pFIn);
54             for (i = 1; i < length+2; i++){
55                 printf("%c", stringp[i]);
56             }

```

```

56         }
57
58     }
59 }
60 }
61 free(stringp);
62 return stringp;
63
64 }
65

```

## fgsmain.c source code

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include "Strings.h"
4  int main(int argc, char* argv[])
5  {
6      FILE *f = fopen("q2input.txt", "r");
7      if (f == (FILE*)NULL)
8      {
9          printf("Error opening input.txt\n");
10     }
11     fgetString(f);
12     free(f);
13 }
14
```

## q2input.txt

---

```
9 I love C!
6 Hello!
6 Hoasdp|
```



## Makefile

```
1 Strings.o: Strings.c Strings.h
2 gcc -c Strings.c
3 main: main.o Strings.o
4 gcc -o main main.o Strings.o
5 main.o: main.c
6 gcc -c main.c
7 run: main
8 ./main output.txt
9 fgsmain.o: fgsmain.c
10 gcc -c fgsmain.c
11 fgsmain: fgsmain.o Strings.o
12 gcc -o fgsmain fgsmain.o Strings.o
13 run2: fgsmain
14 ./fgsmain
15 |
```

## Terminal output

```
[mmoustaf@gcl12m38 Lab4]$ ./fgsmain
I love C!
Hello!
Hoasdp
```

---

## Cloning git

```
[mmoustaf@gcl12m38 Lab4]$ git clone https://vcs.cs.unb.ca/git/cs2263-mmoustaf
Cloning into 'cs2263-mmoustaf'...
Username for 'https://vcs.cs.unb.ca': mmoustaf
Password for 'https://mmoustaf@vcs.cs.unb.ca':
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (12/12), done.
[mmoustaf@gcl12m38 Lab4]$ cd
cs2263-mmoustaf/ input.txt      Makefile      Strings.c
fgsmain*          main*         output.txt    Strings.h
fgsmain.c         main.c       q2input.txt   Strings.o
fgsmain.o         main.o       q2output.txt
[mmoustaf@gcl12m38 Lab4]$ cd cs2263-mmoustaf/
[mmoustaf@gcl12m38 cs2263-mmoustaf]$ █
```

## Adding to git

```
[mmoustaf@gc112m38 cs2263-mmoustaf]$ git add fgsmain.c main.c Strings.c Strings.h
```

## Committing

```
[mmoustaf@gc112m38 cs2263-mmoustaf]$ git commit -m "Adding"
[master b665519] Adding
 4 files changed, 124 insertions(+)
 create mode 100644 Strings.c
 create mode 100644 Strings.h
 create mode 100644 fgsmain.c
 create mode 100644 main.c
```

## Pushing

```
[mmoustaf@gc112m38 cs2263-mmoustaf]$ git push origin master
Username for 'https://vcs.cs.unb.ca': mmoustaf
Password for 'https://mmoustaf@vcs.cs.unb.ca':
Counting objects: 7, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.55 KiB | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://vcs.cs.unb.ca/git/cs2263-mmoustaf
 8b82a53..b665519  master -> master_
```

## Makefile

```
1 Strings.o: Strings.c Strings.h
2     gcc -c Strings.c
3 main: main.o Strings.o
4     gcc -o main main.o Strings.o
5 main.o: main.c
6     gcc -c main.c
7 run: main
8     ./main output.txt
9 fgsmain.o: fgsmain.c
10    gcc -c fgsmain.c
11 fgsmain: fgsmain.o Strings.o
12    gcc -o fgsmain fgsmain.o Strings.o
13 run2: fgsmain
14    ./fgsmain
```