FND18

Strings.h

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

#ifndef STRINGS_H
#define STRINGS_H
typedef char *String;
typedef char** StringList;
// 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);
StringList duplicateStringList(int i, StringList 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;
}
StringList duplicateStringList(int i, StringList sl)
{
    StringList copy = (StringList )malloc(sizeof(String) * i);
    int j;
    for (j = 0; j < i; j++)
    {
        copy[j] = sl[j];
    }
```

```
37      return copy;
38  }
39  int compareStrings(void *s1, void *s2)
40  {
41      StringList sc1 = (String*)s1;
42      StringList 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
```

test.c

```
1   #include "Strings.h"
2   #include <stdlib.h>
3   #include <stdio.h>
4   int main(int argc, StringList argv)
5   {
6       StringList copy = duplicateStringList(argc, argv);
7       int i;
8       for(i = 0; i < argc; i++){
9           printf("%s\n", copy[i]);
10      }
11      return EXIT_SUCCESS;
12  }
13
```

Testing the program

```
[mmoustaf@gc112m38 FND18]$ gcc -o test test.c Strings.c
[mmoustaf@gc112m38 FND18]$ ./test
./test
[mmoustaf@gc112m38 FND18]$ ./test Hello My name is Mahmoud
./test
Hello
My
name
is
Mahmoud
[mmoustaf@gc112m38 FND18]$
```