Exercise 1:

The header file `mystr.h` and the program `test.c` are given.

mystr.h

```
#if !defined MYSTR
#define MYSTR
/* calculate the length of a
string */
int slen(char*);

/* concatenate two strings
into a new string */
char* scat(char*, char*);

/* compares two strings s1
and s2, returns an integer
less than, equal to, or
greater than zero if s1 is
found, respectively, to be
less than, to match, or be
greater than s2 */
int scmp(char*, char*);

/* returns a new duplicate
of the parameter */
char* sdup(char*);

/* returns a new string that
is the reverse of the
parameter */
char* srev(char*);

#endif
```

test.c

```
#include <stdio.h>
#include <stdlib.h>
#include "mystr.h"
int main(){
  char name1[20], name2[20];
  printf("Enter your name: ");
  scanf("%s", name1);

  char* phrase = scat("Hello ", name1);
  printf("%s\n", phrase);
  free(phrase);
  printf("Enter your name: ");
  scanf("%s", name2);

  if (!scmp(name1, name2)){
    phrase = scat("You already entered ", name2);
    printf("%s\n", phrase);
  }
  else {
    phrase = scat("Hello ", name2);
    printf("%s\n", phrase);
  }

  char* dupname = sdup(name1);
  printf("%s\n", dupname);

  char* revname = srev(name2);
  printf("%s\n", revname);
  return 0;
}
```

1. **Launch the terminal**
2. **Create a new directory with the name "Lab05" inside "CSC215"**
3. **Implement the header file in source file `mystr.c` , taking in consideration that:**
   a. **All array traversing is done using pointer arithmetic.**
   b. **All new strings are created dynamically.**
4. **You can always test your code against compilation errors using gcc.** **7 points**

Answer:

```
// mystr.c

#include <stdlib.h>
#include "mystr.h"

int slen(char* str){
 int i=0;
 while(*str++) i++;
 return i;
}

char* scat(char* s1, char* s2){
 char *c, *cat = (char*)calloc(slen(s1)+slen(s2)+1, 1);
 if ((c=cat)){
  while((*c++=*s1++));
  c--;
  while((*c++=*s2++));
 }
 return cat;
}

int scmp(char* s1, char* s2){
 while(*s1 && (*s1==*s2))
  s1++, s2++;
 return *s1-*s2;

}

char* sdup(char* str){
 char *d, *dup = (char*)calloc(slen(str)+1, 1);
 if ((d=dup))
  while((*d++=*str++));
 return dup;
}

char* srev(char* str){
```

```
 char *rs, *re, *rev = (char*)calloc(slen(str)+1, 1);
 if ((rs=rev)) {
  re = str+slen(str)-1;
  while (re >= str)
    *rs++ = *re--;
 }

 return rev;

}
```

****

```
//mystr.h
#if !defined MYSTR
#define MYSTR
/* calculate the length of a string */
int slen(char*);

/* concatenate two strings into a new string */
char* scat(char*, char*);

/* compares two strings s1 and s2, returns an integer less than, equal to, or greater than zero if s1 is found, respectively, to be less than, to match, or be greater than s2 */
int scmp(char*, char*);

/* returns a new duplicate of the parameter */
char* sdup(char*);

/* returns a new string that is the reverse of the parameter */
char* srev(char*);

#endif
```

```
//test.c

#include <stdio.h>
#include <stdlib.h>
#include "mystr.h"
int main(){
  char name1[20], name2[20];
  printf("Enter your name: ");
```

```
  scanf("%s", name1);

  char* phrase = scat("Hello ", name1);
  printf("%s\n", phrase);
  free(phrase);
  printf("Enter your name: ");
  scanf("%s", name2);

  if (!scmp(name1, name2)){
    phrase = scat("You already entered ", name2);
    printf("%s\n", phrase);
  }
  else {
    phrase = scat("Hello ", name2);
    printf("%s\n", phrase);
  }

  char* dupname = sdup(name1);
  printf("%s\n", dupname);

  char* revname = srev(name2);
  printf("%s\n", revname);
  return 0;
}
```

Output :

```
vanarP

C:\Users\ppathak\Documents>gcc -Wall -ansi -o output test.c mystr.c

C:\Users\ppathak\Documents>output
Enter your name: new
Hello new
Enter your name: avtar
Hello avtar
new
ratva

C:\Users\ppathak\Documents>
```

Lab assignment: **3 points**

Write a C program assignment.c that asks the user to enter number of courses then reads the student's grade in each course and finally .calculates and prints the average grade of all courses. The program should store the grades in a dynamically allocated array and visit its elements using pointer arithmetic.

Sample run

```
Enter the number of courses: 4
Enter the grade of Course# 1:88
Enter the grade of Course# 2:98
Enter the grade of Course# 3:100
Enter the grade of Course# 4:90
The average of array elements : 94.00
```