The `malloc()` function reserves a block of memory of the specified number of bytes. And, it returns a pointer of void which can be casted into pointers of any form.

```
void *malloc(size_t size);
```

- ★ malloc function is used to allocate space in memory during program execution.
- ★ malloc function does not initialize the memory allocated during execution.
- ★ malloc function returns null pointer if it cannot allocate requested amount of memory.

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(const char*);

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

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

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

Lab assignment:

3 points

Write a C program assignment.c that asks the user to enter a number of integer values to process. The program then should read these values from the keyboard and assure each of them is unique, and finally print their average. The program should store the grades in a dynamically allocated array and visit its elements using pointer arithmetic.

Sample run

```
Enter the number of integers to process: 4
Enter value # 1: 88
Enter value # 2: 98
Enter value # 3: 88
You have entered 88 already
Enter value # 3: 75
Enter value # 4: 90
The average of array elements : 87.75
```