KSU/CCIS/CS	**CSC 215**	**Mid-term exam 2 - Fall 13-14** **Time allowed: 1:30**
Name: .... ID:		

EXERCISE 1

Write True/ False   (14pts)

A local variable is one whose value can be accessed only by the Function/block in which it is declared.	
Calling **free** on the same address twice is ok.	
strlen(s) returns the number of characters in s including the terminating character.	
The value returned by **isalnum('a')** is 1	
The unary operators & and * have the same precedence as any other unary operator, with associativity from right to left	
A variable is of **static** storage class if a cell is allocated to it upon entry to a segment of code and deallocated upon exit from this segment.	
A variable is of **automatic** storage class if a cell is allocated to it at the beginning of the program execution and remains allocated until the program execution terminates	

EXERCISE 2

Select the correct answer   (10pts)

1. Which of the following is the proper keyword or function to allocate memory in C?
    a. new
    b. malloc
    c. create
    d. allocate

2. Which of the following is the proper keyword or function to deallocate memory?
    A. free
    B. delete
    C. clear
    D. dealocate

3. Which of the following differences between malloc and calloc **is not** true?

   A. malloc allocates number of bytes passed as argument
   B. calloc allocates the product of number of elements multiplied by the size of each element, which are both passed as arguments.
   C. both malloc and calloc return void*
   D. both malloc and calloc initialize allocated memory to all 0

4. What gets printed by the code below? (Assume 1 byte characters)

```
char array[] = "foo";

printf("%lu\n", sizeof(array[0]));
```

   A. 0
   B. 1
   C. 2
   D. f

5. When a function calls itself (directly, or indirectly) it is called a

   A. Self
   B. Recursive
   C. Referring
   D. None of the above

EXERCISE 3

1- Write the code to create an array of 10 integers and dynamically allocate the memory to the elements of the array.   (6 pts)

2- Write the function **int isdigit(int c)**   (10pts)

3- Write the function **int toupper(int c)** (10pts)

(you may use functions in the ctype library other than toupper)

4- Write the output of the following C program. (8 pts)

```c
#include <stdio.h>
int main()
{
   char s[100] = "riyadh";
   char *p1 = &s[1];
   printf("The value of *p1 is %c", *p1);
   printf("The value of *p2 is %c", *++p1);
   char *p2 = &s[3];
   printf("The value of p2-p1 is %d", p2 - p1);
   printf("The value of p1-p2 %d", p1 - p2);
   return 0;
}
```

5- Write the output of the corresponding segment of code (5pts)

```c
const char haystack[25] = "I love my csc215 course.";
const char needle[4] = "215";
char *ret;
ret = strstr(haystack, needle);
printf("The substring is: %s\n", ret);
```

6- Write the output of the corresponding C program   (5 pts)

```c
#include <stdio.h>

void printSeries(int num) {
    if (num > 1)
        printSeries(num - 1);
    printf("%d\n", num);
}
main()
{
    printSeries(5);
}
```

7- Write the output of the corresponding segment of code   (2pts)

```c
char str1[10],str2[10];
 int ret;

 strcpy(str1, "abcdef");
 strcpy(str2, "ABCDEF");

 ret = strcmp(str1, str2);

if(ret > 0)
   printf("str1 is less than str2");
else if(ret < 0)
   printf("str2 is less than str1");
else
   printf("str1 is equal to str2");
```

EXERCISE 5

Write a C program that implements the following requirements: (30pts)

1- A function called **max** that takes two integers and return the maximum.
2- A recursive function called **sum** that takes an integer n and returns the sum from 1 to n.
   (e.g: sum(5) = 1+2+3+4+5)
3- A recursive function called **prod** that takes an integer n and returns the product from 1 to n.
   (e.g: prod(5) = 1x2x3x4x5)
4- A **main** function with the following requirements:
   a. Ask the user to enter two numbers and read them one at time.
   b. Compute the maximum of the two numbers and save it into a variable called **m**.
   c. Print the running sum of **m**.
   d. Print the running product of **m**.