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

EXERCISE 1

Write True/ False (20pts)

In C, **boolean** is the logical type	
In C, **memory management** is left to the programmer.	
C helps **organize** software projects more than Java.	
The conversion of a **higher** order type to a **lower** order may cause truncation and loss of information.	
The scope of a **global** variable is the entire program.	
strlen(s) returns the number of characters in s including the terminating character.	
A **local** variable is one whose value can be accessed only by the Function/block in which it is declared.	
The operator **&**, when applied to a variable, results in the address of the variable.	
Pointers of different types have **different** sizes.	
The **continue** statement does not terminate the loop; it only interrupts a particular iteration.	

EXERCISE 2

Select the correct answer (20pts)

1- Which of the following is **NOT** a correct for naming variables in C?

a) May begin with a letter
b) Cannot contain white space characters
c) Cannot begin with an underscore
d) Must not be a keyword

2- What is 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

3- Given the following declaration **int i=1, *ip;** Which of the flowing initializes the pointer ip to the address of i?

e) ip = &i; b) *ip = i; c) i = &ip; d) *ip=&i;

4- When a break statement is encountered within a loop body,

a) The execution of the loop body is interrupted, and the program control transfers to the exit point of the loop.
b) All the remaining statements in the loop body are skipped and the loop continuation condition is evaluated next.
c) The program stops.
d) Nothing happens.

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 output of the following C program. (10 pts)

```
#include <stdio.h>
void main()
   int a = 2 , b=3, c=4;

   int *p = &a;

   printf("a and *p: %d %d\n", a, *p);

   (*p) +=1;

   printf("a and *p: %d %d\n", a, *p);

   printf("a > b: %d\n", a>b);

   printf("a-c==b+c : %d\n", a+c==b+c);

   printf("c<<2: %d\n" , c<<2);
}
```

2- Write the output of the following C program. (10pts)

```
#include <stdio.h>
int main()

   int i, n=10, sum=0;
   for (i = 1; i <= n; i++) {
        if (i % 3 == 0) { continue; }
        sum += i;
   }
   printf("The value of sum is %d\n", sum);

   sum=0;
   for (i = 1; i <= n; i++) {
        if (i % 4 == 0) { break; }
        sum += i;
   }
   printf("The value of sum is %d\n", sum);

   sum=0;
   while(sum<=n){
      sum++;
   }
   printf("The value of sum is %d\n", sum);

   return 0;
}
```

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

```
#include <stdio.h>

void printSeries(int num) {
   if (num > 1)
      printSeries(num - 1);
   printf("%d\n", num);
}
main()
{
   printSeries(4);
}
```

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

```
float x = 10;
void doubleX()
{
       x *=2;
       printf("%f", x);
}
main(){
       float x = 3;
       doubleX();
       printf("The value of x is: %f", x);
}
```

EXERCISE 3

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

1- A function called **max** that takes two integers and return their maximum.
2- A recursive function called **factorial** that takes an integer n and returns the factorial of n. (e.g: factorial(5) = 5*4*3*2*1=120)
3- A **main** function with the following requirements:
    a. Ask the user to enter two numbers and read them **one at time** using **scanf**.
    b. Compute the maximum of the two numbers using the function **max** and save the result into a variable called **m**.
    c. Compute the factorial of m using the factorial function and save the result into a variable called **f**.
    d. Print the value of **f**.