Name: Hemos ID:

Student ID (학번):

CSED-101 Introduction to Computing, Spring 2010
Midterm

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Total
Your Score												
Max Score	9	4	10	12	10	8	9	5	12	9	12	100

- Write down your name, hemos ID, and student ID.
- There are 13 pages in this midterm.
- Your answers must run correctly in C programming language without error or warning. Otherwise, your answers will be considered incorrect. For example, it is ok to put more parentheses than needed in your answer, but it will be incorrect if you put fewer parentheses than needed.
- You must write your answer on the underline => _____. Scratches outside the underline will be ignored.
- The total score is 100.
- This is a 3-hour exam.

1. (9 points) A correct answer will be given one point. However, <u>an incorrect answer will be given a −1 point</u>. A blank answer will be given a zero point. Mark T if the statement is true, otherwise mark F.

(1) [<u>F</u>]   Loader is responsible for making an executable file by merging object codes and necessary libraries.

(2) [<u>F</u>]   All C variables must start with an alphabetic character.

(3) [<u>T</u>]   Assume that z = (x == y) ? 1 : 0. If x = 3 and y = 3, then z = 1.

(4) [<u>F</u>]   Given x = 5, y = x+++1 sets y to 7.

(5) [<u>T</u>]   The declared variables in a function are local variables.

(6) [<u>F</u>]   Automatic variables are initialized to zero automatically.

(7) [<u>F</u>]   The break statement is always required in the switch selection structure.

(8) [<u>F</u>]   The **for** statement cannot be used for an infinite loop but the **while** statement can.

(9) [<u>T</u>]   A function may return at most one value.

2. (4 pts)Answer the following questions.

2.1. (2 pts)The following program generates a random integer in the range 1 to 6. To complete the program, fill out the blanks.

```c
#include <stdio.h>
#include <stdlib.h> /* header file for rand function */
#include <time.h>
int main()
{
    int x;
    srand(time(NULL));
    x = [rand() % 6 + 1 ];
    printf("%d\n", x);
    return 0;
}
```

2.2. (2 pts) The following program generates a random real numbers in the range 1 to 2. To complete the program, fill out the blanks.

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    float x;
    srand(time(NULL));
    x = [ (float)rand() / RAND_MAX + 1 ];
    printf("%f\n", x);
    return 0;
}
```

[채점 기준]
- (float) 없으면 1 점
- 1/rand(), 1/(rand()+1) -> divided by zero 문제 발생 (0 점)

3. (10 pts) Write a program that splits one-, ten-, and hundred-digit for a given number *n*. We assume the number is between 0 and 999.

For example, the output of the following program should be

    123 = 3*1 + 2*10 + 1*100
    123 = 3*1 + 2*10 + 1*100

```c
int main()
{
    int one, ten, hundred;
    int n = 123;

    split(n, &one, &ten, &hundred);

    printf("%d = %d*1 + %d*10 + %d*100\n", n, one, ten, hundred);
    printf("%d = %d*1 + %d*10 + %d*100\n", n, split_one(n), split_ten(n), split_hundred(n));

    return 0;
}
```

In other words, write the source code for <u>the following user-defined functions</u> in c programming language.

```c
/* (4 pts) split(n, &one, &ten, &hundred) */
void split(int n, int* one, int* ten, int* hundred)
{
    *one = n%10;
    *ten = n/10%10;
    *hundred = n/100%10;   (or *hundred = n/100; or return (n%1000)/100;))
}
```

[채점기준]
- 함수 헤더 부분 틀리면 0점

```c
/* (2 pts) split_one(n) */
int split_one(int n)
{
    return n % 10;
}
```

```c
/* (2 pts) split_ten(n) */
int split_ten(int n)
{
    return n/10%10; (or return (n % 100) / 10;)
}
```

```c
/* (2 pts) split_hundred(n)*/
int split_hundred(int n)
{
    return n/100%10;   (or return = n/100; or return (n%1000)/100;)
}
```

4. (12 pts) This is a program which prints four right triangles below.

```
  *   *
 **   **
***   ***
****   ****
*****   *****

*****   *****
****   ****
***   ***
**   **
*   *
```

To complete the program, fill out the blanks. (Remind the assign 3)

```c
#include <stdio.h>

void main()
{
        int i, j;

        for(i=0; i<11; i++) {
                if(i<5)  {
                        for(j=0; j<5; j++) {
                                if(  ①  )
                                        putch('*');
                                else
                                        putch(' ');
                        }
                        putch(' ');
                        for(j=0; j<5; j++) {
                                if(  ②  )
                                        putch('*');
                                else
                                        putch(' ');
                        }
                        putch('\n');
                }
                else if(i>5) {
                        for(j=0; j<5; j++) {
                                if(  ③  )
                                        putch('*');
                                else
                                        putch(' ');
                        }
                        putch(' ');
                        for(j=0; j<5; j++) {
                                if(  ④  )
                                        putch('*');
                                else
                                        putch(' ');
                        }
                        putch('\n');
                }
                else
                        putch('\n');
        }
}
```

Answers:

① (3 pts) i+j>3   or i+j >=4

② (3 pts) i>=j   or i−j >=0

③ (3 pts) i−j<7   or i-j <= 6

④ (3 pts) j+i<11   or i+j <=10

[채점기준]
- 올바르게 실행되면 맞음.
- ≥ 이런식으로 쓴 경우 : -1점

5. (10 pts) This is a program using <u>recursion</u>. The program receives a positive integer as an input and converts it into the corresponding binary number. Fill out the blanks.

Example)

Input the number: 10
The decimal number 10 is represented in binary as 1010.
Input the number: 22
The decimal number 22 is represented in binary as 10110.

```c
#include<stdio.h>
int dec2bin(int n);

int main(){

    int number;
    int result;

    printf("Input the number: ");
    scanf("%d", &number);

    result = dec2bin(number);
    printf("The decimal number %d is represented in binary as %d.\n", number, result);

    return 0;
}

int dec2bin(int n)
{
    if( ____①____ )
        return n;
    else
        return _____②_____;
}
```

Answers:

① (3pts) __n < 2   or   n==1 등__

② (7pts) (n % 2) + 10 * dec2bin(n/2)

6. (8 pts) This is a simple code snippet which increases or decreases the variables using prefix, postfix, and logic operations. What is the output of the code?

```
int a, b, c;

a = 0; b = 1; c = 1;

printf("0. a=%d, b=%d, c=%d\n", a, b, c);

if(a||b)
        printf("1. a=%d, b=%d, c=%d\n", a, b, c--);

if(c&&b++)
        printf("2. a=%d, b=%d, c=%d\n", a++, b, c);

if(++a&&b)
        printf("3. a=%d, b=%d, c=%d\n", a, b, ++c);

printf("4. a=%d, b=%d, c=%d\n", a, b, c);
```

Answers:

0	.		a	=	0	,		b	=	1	,		c	=	1				
1	.		a	=	0	,		b	=	1	,		c	=	1				
3	.		a	=	1	,		b	=	1	,		c	=	1				
4	.		a	=	1	,		b	=	1	,		c	=	1				

[채점기준]
다 맞으면 8점
맨 왼쪽컬럼 1 3 4(순서)를 틀리거나 2번이 나오면 –3점
나머지는 한 개 틀릴때마다 –1점
따라서 모두 틀렸을 경우 8 – 3 – 9 = –4 점

7. (9 pts) Answer the following question.

```
char grade(float score, int absence, int tardy) {
    absence += tardy/3;
    if(absence<3) {
        if(score>=90.0) {
            return 'A';
        }
        else if(score>=80.0) {
            return 'B';
        }
        else if(score>=70.0) {
            return 'C';
        }
        else if(score>=60.0) {
            return 'D';
        }
        else {
            return 'F';
        }
    }
    else {
        return 'F';
    }
}
```

Change above nested **if** into **case**. Fill out the blanks ①~③.

```
char grade(float score, int absence, int tardy) {
    absence += tardy/3;
    switch ( ___①___ ) {
        case 0: break;
        default: return 'F';
    }
    switch ( ___②___ ) {
        case ③: return 'A';
        case 8: return 'B';
        case 7: return 'C';
        case 6: return 'D';
        default: return 'F';
    }
}
```

Answers:

① (4점) (int)(absence/3)

② (4점) (int)score/10

③ (1점) 9

[채점기준]

① (int)없어도 정답, absence>=3 등도 인정

② (int)없으면 2점

8. (5 pts) Complete the table with the results when we open a file using
FILE* fp = fopen("infile.dat", "___"); statement with three different modes (r, w and a). (You can use Korean for the answer)

	r	w	a
If the infile.dat file exists	File을 읽기 위해 연다.	File을 쓰기 위해 새로운 infile.dat를 연다. (기존의 내용은 사라짐)	infile에 내용을 추가하기 위해 파일을 연다.
If the infile.dat file doesn't exist	File을 열지 못하고 NULL 리턴	File을 쓰기 위해 infile.dat를 생성한다.	File을 쓰기 위해 infile.dat를 생성한다.

[채점기준] 1개당 1점씩

9. (12 pts) Write a program which gets original and target file names from the user and copies whole contents of original file into the target file. (Assumption: the contents of the original file are written in text format.)

```c
/* Copy the contents of one file into another */
#include <stdio.h>
void copy_file(FILE *original, FILE *copy);

void main()
{
    char filename[FILENAME_MAX];
    FILE *infile;
    FILE *outfile;
    printf("This program copies files.\n");
    printf("Origianl file: ");
    scanf("%s", filename);
    infile = fopen(filename, "r");

    if(   infile == NULL        ) printf("Cannot open %s\n", filename);
    else {
        printf("Copy: ");
        scanf("%s", filename);
        outfile = fopen(filename, "w");
        copy_file(infile, outfile);

        fclose(infile);
        fclose(outfile);
    }
}
```

```
void copy_file(FILE *original, FILE *copy)
{
    char next_char;
    int flag;
    flag = fscanf(original, "%c", &next_char);
    while (flag != EOF){
        fprintf(copy, "%c", next_char);
        flag = fscanf(original, "%c", &next_char);
    }
}
```

[채점기준]
① main()의 빈칸은 각 0.5점
② copy_file(FILE *original, FILE *copy) 7점
   함수 헤더 부분 틀린 경우 -2점

10. (9 pts) Complete the <u>recursive</u> function **sum(int n, int m)**, which computes the sum of integers from n to m. For example, sum(3, 6) returns 18.

```
int sum (int n, int m)
{
    if( _____①_____ )
        return _____②_____;
    else
        return _____③_____;
}
```

Answers:

① (3pts) ___ n == m ___ or ___ n <= m 등 (n=m 은 틀림, ≤ 등의 경우 -1점) _____

② (3pts) ___ n (or m) _____

③ (3pts) ___ n + sum(n+1, m) (or m + sum(n, m-1)) _____

11. (12 pts) Complete the <u>recursive</u> function **seq(int i, int* p, int* q)**, which computes the following sequence a_n (=p) and b_n (=q)

$a_n = 2a_{n-1} + b_{n-1}$

$b_n = b_{n-1} + a_{n-1}$

$a_1 = 2$

$b_1 = 1$

For example, the main function will display the results as follows.

i=1, p=2, q=1

i=2, p=5, q=3

i=3, p=13, q=8

i=4, p=34, q=21

```c
#include <stdio.h>
void seq(int i, int *p, int *q);

int main()
{
        int i, p, q;
        for (i=1; i<=4; i++)
        {
                p=0; q=0; //initialize p & q values
                seq(i, &p, &q);
                printf("i=%d, p=%d, q=%d\n", i, p, q);
        }
        return 0;
}

void seq(int i, int* p, int* q)
{
        int pcopy;
        int qcopy;
        if (i > 1)
        {
                seq(i-1, &pcopy, &qcopy);
                *p = 2*pcopy + qcopy;
                *q = pcopy + qcopy;
                return;
        }
        else
        {
                *p = 2;
                *q = 1;
                return;
        }
}
```

[채점기준]

- 변수명은 다를 수 있음. 프로그램이 맞게 돌아가면 12점. 아닌 경우 0점

- 함수명 안 쓰고 호출한 경우 -6점