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, an incorrect answer will be given a –1 point. A blank answer will be given a zero point. Mark T if the statement is true, otherwise mark F.

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

(2) [ ] All C variables must start with an alphabetic character.

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

(4) [ ] Given x = 5, y = x+++1 sets y to 7.

(5) [ ] The declared variables in a function are local variables.

(6) [ ] Automatic variables are initialized to zero automatically.

(7) [ ] The break statement is always required in the switch selection structure.

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

(9) [ ] 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.

```
#include <stdio.h>
#include <stdlib.h> /* header file for rand function */
#include <time.h>
int main()
{
	int x;
	srand(time(NULL));
	x = [                    ];
	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.

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

int main()
{
	float x;
	srand(time(NULL));
	x = [                                    ];
	printf("%f₩n", x);
	return 0;
 }
```

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

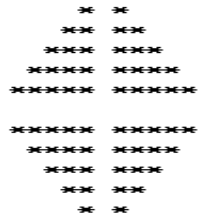
123 = 3*1 + 2*10 + 1*100

```
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 the following user-defined functions in c programming language.

/* (4 pts) split(n, &one, &ten, &hundred) */
/* (2 pts) split_one(n) */
/* (2 pts) split_ten(n) */
/* (2 pts) split_hundred(n)*/

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)

```
#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) _____

② (3 pts) _____

③ (3 pts) _____

④ (3 pts) _____

5. (10 pts) This is a program using recursion. 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.
```

```
#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) _____

② (7pts) _____

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**			

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) _____

② (4점) _____

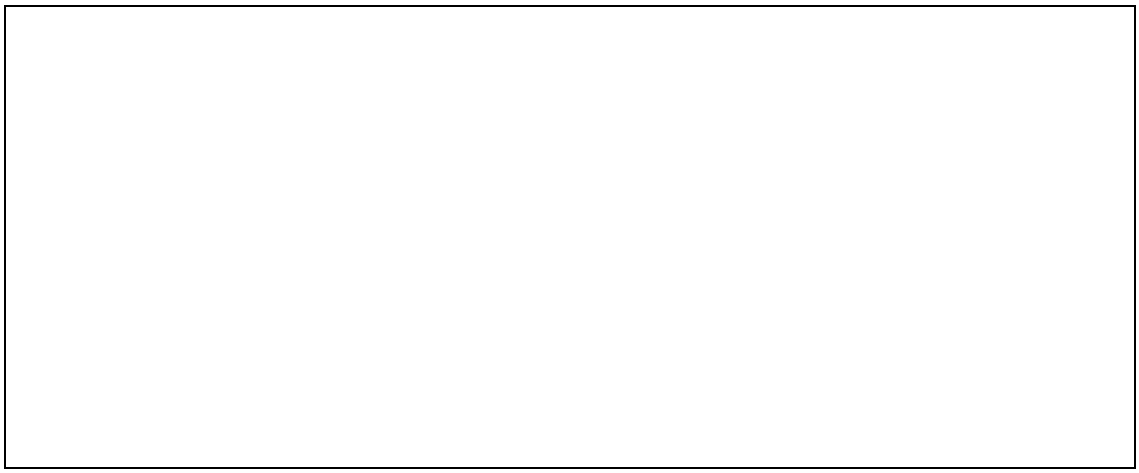
③ (1점) _____

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.		
If the infile.dat file doesn't exist			

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.)

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

void main()
{
    char filename[FILENAME_MAX];
    _____infile;
    _____outfile;
    printf("This program copies files.₩n");
    printf("Origianl file: ");
    scanf("%s", filename);
    ____________________________

    if( ____________________ ) printf("Cannot open %s₩n", filename);
    else {
        printf("Copy: ");
        scanf("%s", filename);
        ____________________________
        copy_file(infile, outfile);

        ___________
        ___________
    }
}
```



10. (9 pts) Complete the recursive 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) _____

② (3pts) _____

③ (3pts) _____

11. (12 pts) Complete the recursive 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

```
#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)
{


	if (i > 1)
	{



		return;
	}
	else
	{



		return;
	}
}
```