

프로그래밍언어 (실습)

**실습 10 (보충설명) - 문자열 (string),  
파일 입출력, 이진파일, 암호화**



**교수 김 영 탁**

**영남대학교 정보통신공학과**

(Tel : +82-53-810-2497; E-mail : ytkim@yu.ac.kr)

# Outline

- ◆ 문자열의 파일 입력, 정렬 및 파일 출력
- ◆ 학생 데이터 레코드의 랜덤 파일 입출력
- ◆ 이진파일의 **dump**
- ◆ 간단한 문자열의 16진수 문자로의 암호화
- ◆ **main()** 함수



**문자열의 파일 입력, 정렬 및 파일 출력**

# 문자 (character) 단위 입출력 라이브러리

입출력 함수	설명
int getchar(void)	하나의 문자를 읽어서 반환한다.
void putchar(int c)	변수 c에 저장된 문자를 출력한다.
int getch(void)	하나의 문자를 읽어서 반환한다(버퍼를 사용하지 않음).
void putch(int c)	변수 c에 저장된 문자를 출력한다(버퍼를 사용하지 않음).
scanf("%c", &c)	하나의 문자를 읽어서 변수 c에 저장한다.
printf("%c", c);	변수 c에 저장된 문자를 출력한다.



# 문자열 (string, word) 단위 입출력 라이브러리 함수

입출력 함수	설명
int scanf("%s", s)	문자열을 읽어서 문자배열 s[]에 저장
int printf("%s", s)	배열 s[]에 저장되어 있는 문자열을 출력한다.
char *gets(char *s)	한 줄의 문자열을 읽어서 문자 배열 s[]에 저장한다.
int puts(const char *s)	배열 s[]에 저장되어 있는 한 줄의 문자열을 출력한다.



# 문자열 처리 라이브러리

문자열 처리 함수	함수 설명
<code>int strlen(char *str);</code>	문자열 <code>str</code> 의 길이를 반환
<code>strcpy(char *s1, char *s2);</code>	문자열 <code>s2</code> 를 <code>s1</code> 으로 복사
<code>strncpy(char *s1, char *s2, int n);</code>	문자열 <code>s1</code> 과 <code>s2</code> 를 최대 <code>n</code> 개 문자까지 복사
<code>strcat(char *s1, char *s2);</code>	문자열 <code>s2</code> 를 <code>s1</code> 에 이어 붙임 (concatenate)
<code>strncat(char *s1, char *s2, int n);</code>	문자열 <code>s2</code> 의 최대 <code>n</code> 개 문자를 <code>s1</code> 에 이어 붙임 (concatenate)
<code>strcmp(char *s1, char *s2);</code>	문자열 <code>s1</code> 과 <code>s2</code> 를 비교
<code>strncmp(char *s1, char *s2, int n);</code>	문자열 <code>s1</code> 과 <code>s2</code> 를 최대 <code>n</code> 개 문자까지 비교
<code>strchr(char *str, char c);</code>	문자열 <code>str</code> 에 문자 <code>c</code> 가 포함되어 있는가를 확인
<code>strstr(char *s1, char *s2);</code>	문자열 <code>s1</code> 에 문자열 <code>s2</code> 가 포함되어 있는가를 확인



# 문자열을 수치로 변환하는 전용함수

## ◆ 문자열 전용 함수

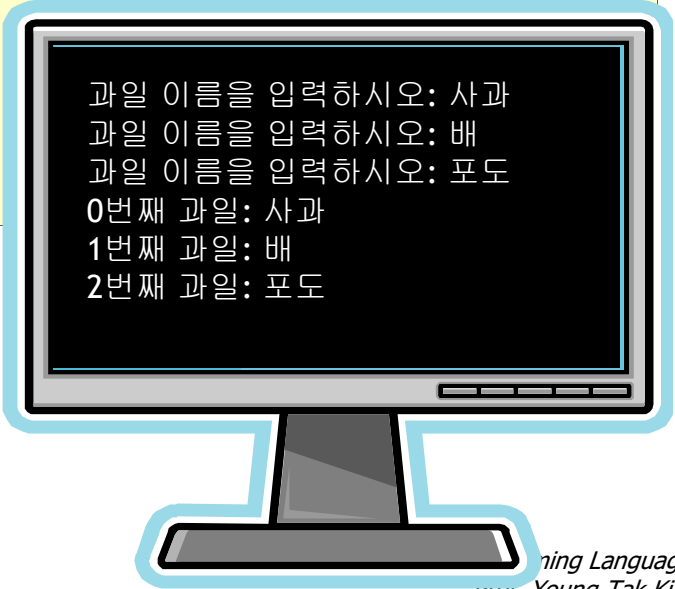
- 다양한 형식을 처리하는 scanf()보다 크기가 작다.
- stdlib.h에 원형 정의 - 반드시 포함

문자열의 수치변환 함수	함수 설명
int atoi(const char *str);	주어진 문자열 str을 정수 데이터로 변환
long atol(const char *str);	주어진 문자열 str을 long 데이터로 변환
double atof(const char *str);	주어진 문자열 str을 더블형 데이터로 변환



# 문자열의 배열을 2차원 문자배열로 구현

```
#include <stdio.h>
#define MAX_STR_LEN 20
int main( void )
{
    int i;
    char fruits[3][MAX_STR_LEN];
    for(i = 0; i < 3; i++) {
        printf("과일 이름을 입력하시오: ");
        scanf("%s", fruits[i]);
    }
    for(i = 0; i < 3; i++)
        printf("%d번째 과일: %s\n", i, fruits[i]);
    return 0;
}
```



과일 이름을 입력하시오: 사과  
과일 이름을 입력하시오: 배  
과일 이름을 입력하시오: 포도  
0번째 과일: 사과  
1번째 과일: 배  
2번째 과일: 포도





## 입력 단어 파일 (words.txt)

expand seek dynamic responsibility generate operational query requirement graduate equivalent familiarity apply completed challenge reply post interest develop maintain specialty resource resume project appreciate expansion sufficient presentation consideration objective summary grade code interactive improve technical fluent hobby interview notice excited opportunity inform congratulation career formal definitely appropriate professional integrate customize client reputation wonderful introduction confidence impression encourage phase semester subject coordination successful patient sensitive optimistic sophisticated participate function member portfolio instructor upload methodology rough accomplish documentation complicated detail coworker receive orientation background mission informative smoothly supervisor direction elevator administration equipment facility provide request permission explanation catalog instruction define fulfill process approval activate reduce interface migrate offer target sector limit host support corporate calendar centralize productive option access device blond shape mention wonder authorization modify electronic progress conference solution cooperate division mainland demand budget capital expenditure environment thought solid storage architecture reconsider discuss rapidly consultant chairperson understanding adopt customer application capability user-friendly communicate download compatible edit share contact conquer dominate alignment benefit assignment perform advice remind submit duty concentrate conduct full-time evolve demand cautious effective achieve refine direct deliver estimate widely expertise specific deadline otherwise delegate motivate exception pick vital engaged expectation handle oriented guidance supervise chair designer perhaps aim agenda establish collaboration participant attendee remotely snapshot contain feature advanced troubleshoot form credit competitor convincing impress referee concerned stage common popular approach breakdown implementation infrastructure respectively gather aspect reasonable duration scenario milestone draft kickoff private government stream module broadcast senior analyst analysis standard procedure variable arrive arrangement schedule brief separate part security confirmation period closely clarify current evaluation suggestion outsource configuration capacity hierarchical logic mapping existing involve deployment somewhat confuse beforehand similar concept neat deliverable specification validate various consist vendor installation critically impact overall promise proposal compile handover latest version up-to-date verify excellent messy gently trial usage commercial practice scale release react initiate bang prevent costly embarrassing victory champion



# 문자열 정렬 – string sorting

```
void selectionSortWords(char **words, int num_words, int max_word_len)
{
    char *temp_word;
    int i, j, min;
    temp_word = (char *)calloc(max_word_len, sizeof(char));
    if (temp_word == NULL)
    {
        printf("Error in dynamic allocation of memory for temp_word !!\n");
        exit;
    }

    /* selection sorting of words */
    for (i = 0; i < num_words; i++)
    {
        strcpy(temp_word, words[i]);
        min = i;
        for (j = i + 1; j < num_words; j++)
        {
            if (strcmp(temp_word, words[j]) > 0)
            {
                min = j;
                strcpy(temp_word, words[j]);
            }
        }
        if (min != i)
        {
            /* temp_word contains words[min] already !! */
            strcpy(words[min], words[i]);
            strcpy(words[i], temp_word);
        }
    }
}
```



**학생 데이터 레코드의 램덤 파일 입출력**

# 랜덤 파일 입출력

## ◆ 랜덤 파일 접근 관련 함수

- fseek() 함수를 사용하여 임의의 파일 위치 (offset로 지정)로 파일 포인터 이동한 후 파일 입력 또는 출력 가능
- rewind()는 파일 포인터를 맨 처음으로 이동시킴
- 구조체로 표현된 data record 단위로 이동

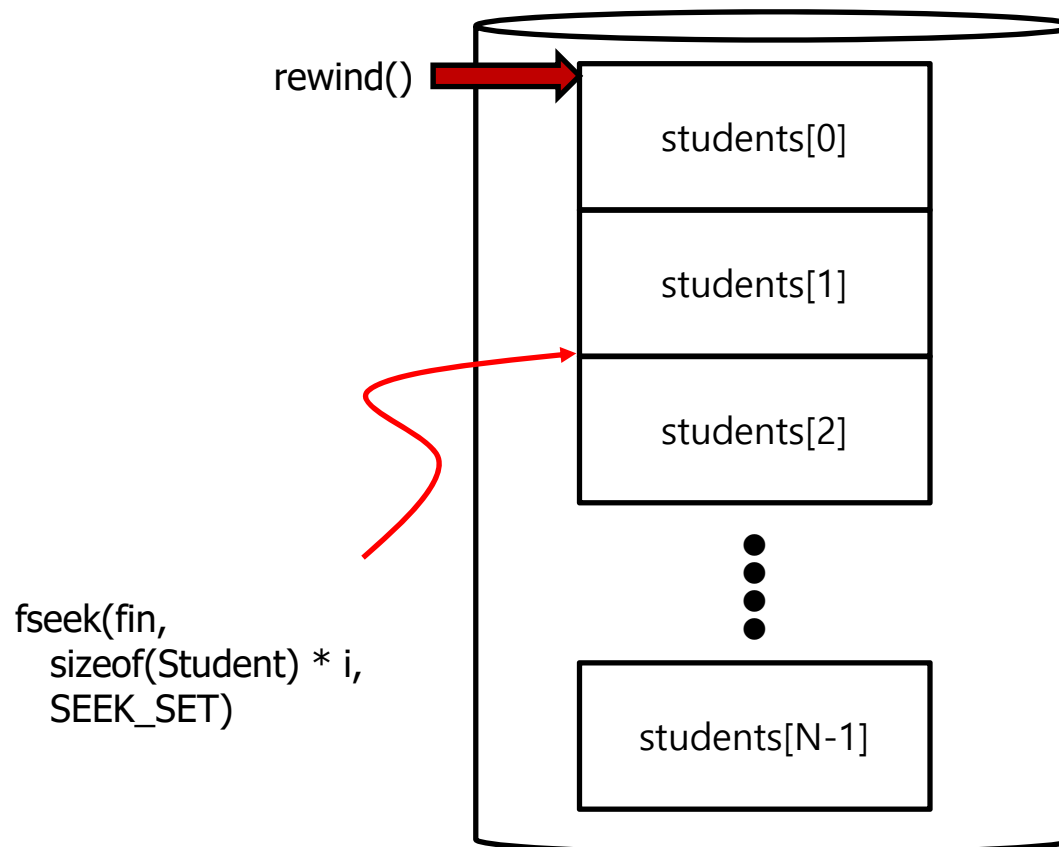
랜덤 파일 접근 관련 함수	의미
int fseek(FILE *fp, long offset, int origin)	지정된 시작위치 (origin)으로 부터 지정된 offset 만큼 이동 Origin으로 설정가능한 Macro: - SEEK_SET : 파일의 처음 - SEEK_END : 파일의 끝 - SEEK_CUR : 현재의 커서 위치
void rewind(FILE *fp)	파일 포인터를 파일의 시작 위치로 이동
long ftell(FILE *fp)	파일 포인터의 현재 위치를 반환
int feof(FILE *fp)	파일의 끝에 도달하였는가를 알려줌. 이진 파일 입출력에서는 주로 사용



# 랜덤 파일 입출력을 위한 파일 포인터 이동

## ◆ `fseek()`, `rewind()`를 사용한 파일 포인터 이동

Random access of file with student records



# 학생 데이터 레코드

```
/* StudentsRecords.cpp */
#include "Student.h"

#define NUM_STUDENTS 10
STUDENT students[NUM_STUDENTS] =
{
    { 21711000, "Kim, G-M", { 1990, 10, 5 }, { 82, 53, 805, 1234 }, 3.57 },
    { 21611075, "Yoon, S-M", { 1990, 4, 5 }, { 82, 53, 811, 1550 }, 4.37 },
    { 21411015, "Hwang, S-S", { 1989, 1, 10 }, { 82, 53, 817, 1005 }, 2.72 },
    { 21611054, "Lee, K-M", { 1991, 5, 15 }, { 82, 10, 9112, 9876 }, 3.35 },
    { 21611340, "Hong, G-M", { 1990, 2, 5 }, { 82, 55, 810, 5678 }, 3.89 },
    { 21712056, "Jang, S-M", { 1990, 3, 15 }, { 82, 10, 9112, 1600 }, 4.42 },
    { 21411017, "Park, S-S", { 1989, 7, 10 }, { 82, 34, 817, 1098 }, 4.12 },
    { 21511053, "Choi, Y-H", { 1992, 9, 25 }, { 82, 53, 845, 5764 }, 3.85 },
    { 21411017, "Shin, D-J", { 1988, 10, 3 }, { 82, 31, 817, 1038 }, 3.21 },
    { 21511053, "Kwak, S-B", { 1994, 11, 15 }, { 82, 2, 897, 8778 }, 4.45 },
};
```



```

/* main.c (1) */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Student.h"

#define NUM_STUDENTS 10
#define MAX_STRING_LEN 512
extern STUDENT students[NUM_STUDENTS];
#define TEST_SORTING
#define TEST_BST_Student

void main()
{
    FILE *fin, *fout;
    STUDENT *pST;
    int record_len;
    char student_record[MAX_STRING_LEN] = { 0 };

    fout = fopen("Sorted_Students.txt", "w");
    if (fout == NULL)
    {
        printf("Error in opening Sorted_Students.txt (write mode)!!\n");
        exit;
    }
    printf("Array of students at initialization : \n");
    printStudents(students, NUM_STUDENTS);
    printf("\n");
}

```



```

/* main.c (2) */

//Sorting students;
selectionSortStudents_by_ST_ID(students, NUM_STUDENTS);
printf("Storing sorted students by increasing order of student ID into
SortedStudent.txt ....\n");
fprintf(fout, students, NUM_STUDENTS);
printStudents(students, NUM_STUDENTS);
fprintf(fout, "\n");
fclose(fout);

fin = fopen("Sorted_Students.txt", "r");
if (fin == NULL)
{
    printf("Error in opening Sorted_Students.txt (read mode)!!\n");
    exit;
}

int cur_pos;
fgets(student_record, MAX_STRING_LEN, fin);
record_len = strlen(student_record);
rewind(fin);
printf("\nRandom access to Sorted_students.txt file (Student record length: %d) ... \n",
record_len);
for (int i = NUM_STUDENTS-1; i >= 0; i--)
{
    fseek(fin, (record_len + 1) * i, SEEK_SET);
    // record_len + 1 to include the CR character at the end of each line
    cur_pos = ftell(fin);
    printf("Current file_position : %3d\n", cur_pos);
    fgets(student_record, MAX_STRING_LEN, fin);
    printf("Student (%2d): %s", i, student_record);
}
fclose(fin);
}

```





```

Array of students at initialization :
Student [ID: 21711000, Kim, G-M , (1990, 10, 5), (tel: +82-053-0805-1234), GPA: 3.57]
Student [ID: 21611075, Yoon, S-M , (1990, 4, 5), (tel: +82-053-0811-1550), GPA: 4.37]
Student [ID: 21411015, Hwang, S-S, (1989, 1, 10), (tel: +82-053-0817-1005), GPA: 2.72]
Student [ID: 21611054, Lee, K-M , (1991, 5, 15), (tel: +82-010-9112-9876), GPA: 3.35]
Student [ID: 21611340, Hong, G-M , (1990, 2, 5), (tel: +82-055-0810-5678), GPA: 3.89]
Student [ID: 21712056, Jang, S-M , (1990, 3, 15), (tel: +82-010-9112-1600), GPA: 4.42]
Student [ID: 21411017, Park, S-S , (1989, 7, 10), (tel: +82-034-0817-1098), GPA: 4.12]
Student [ID: 21511053, Choi, Y-H , (1992, 9, 25), (tel: +82-053-0845-5764), GPA: 3.85]
Student [ID: 21411017, Shin, D-J , (1988, 10, 3), (tel: +82-031-0817-1038), GPA: 3.21]
Student [ID: 21511053, Kwak, S-B , (1994, 11, 15), (tel: +82-002-0897-8778), GPA: 4.45]

Storing sorted students by increasing order of student ID into SortedStudent.txt ....
Student [ID: 21411015, Hwang, S-S, (1989, 1, 10), (tel: +82-053-0817-1005), GPA: 2.72]
Student [ID: 21411017, Park, S-S , (1989, 7, 10), (tel: +82-034-0817-1098), GPA: 4.12]
Student [ID: 21411017, Shin, D-J , (1988, 10, 3), (tel: +82-031-0817-1038), GPA: 3.21]
Student [ID: 21511053, Choi, Y-H , (1992, 9, 25), (tel: +82-053-0845-5764), GPA: 3.85]
Student [ID: 21511053, Kwak, S-B , (1994, 11, 15), (tel: +82-002-0897-8778), GPA: 4.45]
Student [ID: 21611054, Lee, K-M , (1991, 5, 15), (tel: +82-010-9112-9876), GPA: 3.35]
Student [ID: 21611075, Yoon, S-M , (1990, 4, 5), (tel: +82-053-0811-1550), GPA: 4.37]
Student [ID: 21611340, Hong, G-M , (1990, 2, 5), (tel: +82-055-0810-5678), GPA: 3.89]
Student [ID: 21711000, Kim, G-M , (1990, 10, 5), (tel: +82-053-0805-1234), GPA: 3.57]
Student [ID: 21712056, Jang, S-M , (1990, 3, 15), (tel: +82-010-9112-1600), GPA: 4.42]

Random access to Sorted_students.txt file (Student record length: 89) ...
Current file_position : 810
Student ( 9): Student [ID: 21712056, Jang, S-M , (1990, 3, 15), (tel: +82-010-9112-1600), GPA: 4.42]
Current file_position : 720
Student ( 8): Student [ID: 21711000, Kim, G-M , (1990, 10, 5), (tel: +82-053-0805-1234), GPA: 3.57]
Current file_position : 630
Student ( 7): Student [ID: 21611340, Hong, G-M , (1990, 2, 5), (tel: +82-055-0810-5678), GPA: 3.89]
Current file_position : 540
Student ( 6): Student [ID: 21611075, Yoon, S-M , (1990, 4, 5), (tel: +82-053-0811-1550), GPA: 4.37]
Current file_position : 450
Student ( 5): Student [ID: 21611054, Lee, K-M , (1991, 5, 15), (tel: +82-010-9112-9876), GPA: 3.35]
Current file_position : 360
Student ( 4): Student [ID: 21511053, Kwak, S-B , (1994, 11, 15), (tel: +82-002-0897-8778), GPA: 4.45]
Current file_position : 270
Student ( 3): Student [ID: 21511053, Choi, Y-H , (1992, 9, 25), (tel: +82-053-0845-5764), GPA: 3.85]
Current file_position : 180
Student ( 2): Student [ID: 21411017, Shin, D-J , (1988, 10, 3), (tel: +82-031-0817-1038), GPA: 3.21]
Current file_position : 90
Student ( 1): Student [ID: 21411017, Park, S-S , (1989, 7, 10), (tel: +82-034-0817-1098), GPA: 4.12]
Current file_position : 0
Student ( 0): Student [ID: 21411015, Hwang, S-S, (1989, 1, 10), (tel: +82-053-0817-1005), GPA: 2.72]

```



## **이진 파일 (binary file)의 Dump**

# 이진 (Binary) 파일 입출력

## ◆ 이진 파일 (Binary File) 이란?

- 문자열 정보를 저장하는 텍스트파일 (.txt)와는 달리 영상, 음성 등의 멀티미디어 정보 (예: .jpg, .bmp, .mpg, .wma) 또는 프로그램 실행파일 (예: .exe, .lib 등)
- fscanf(), fgets(), fgetc() 함수로 파일을 읽을 수 없음
- 대신, 이진파일 입출력 함수인 fread(), fwrite() 함수를 사용

## ◆ 이진 파일 관련 함수

유형	이진 파일 입출력 함수	설명
이진 데이터 파일 입출력	size_t fread(char *buf, int record_size, int count, FILE *fp)	이진 데이터 파일로부터 record_size 만큼 읽고, 이를 버퍼에 저장
	size_t fwrite(char *buf, int record_size, int count, FILE *fp)	버퍼에 저장되어 있는 데이터 레코드를 이진 파일로 출력
	int feof(FILE *fp)	파일의 끝에 도달하였는가를 알려줌. 이진 파일 입출력에서는 주로 사용됨.



# dumpBinaryFile()

```
#define BUFFER_LEN DUMP_OCTETS_PER_LINE
void dumpBinaryFile(FILE *fin, FILE *fout)
{
    int nbytes;
    unsigned char buffer[BUFFER_LEN] = { '\0' };
    for (int addr = 0; addr < MAX_FILE_SIZE; addr += BUFFER_LEN)
    {
        nbytes = fread(buffer, sizeof(unsigned char), BUFFER_LEN, fin);
        if (nbytes <= 0)
            break;
        fprintf(fout, "%08X: ", addr);
        for (int i = 0; i < nbytes; i++)
        {
            if (i == (BUFFER_LEN / 2))
                fprintf(fout, " ");
            fprintf(fout, "%02X ", buffer[i]);
        }
        fprintf(fout, " ");
        for (int i = 0; i < nbytes; i++)
        {
            if (isprint(buffer[i]))
                fprintf(fout, "%c", buffer[i]);
            else
                fprintf(fout, ".");
        }
        fprintf(fout, "\n");
    }
}
```



```
/* main.c */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include "HandlingBinaryFile.h"
```

```
#define MAX_STR_LEN 100
```

```
void main()
```

```
{
```

```
FILE *fin, *fout;
```

```
char *fname = "ExampleExeCodeFile.exe";
```

```
if ((fin = fopen(fname, "rb")) == NULL)
```

```
{
```

```
printf("Error - binary input file (%s) cannot be opened !!\n", fname);
exit;
```

```
}
```

```
if ((fout = fopen("Output.txt", "w")) == NULL)
```

```
{
```

```
printf("Error - Output.txt cannot be created !!\n");
exit;
```

```
}
```

```
printf("Dumping binary file (%s) ... \n", fname);
```

```
dumpBinaryFile(fin, fout);
```

```
fclose(fin);
```

```
fclose(fout);
```

```
}
```

```
00000000: 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 MZ.....
00000010: B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....@.....
00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 F0 00 00 00 .....
00000040: 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 .....!...L!Th
00000050: 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F is program canno
00000060: 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 t be run in DOS
00000070: 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 mode....S.....
00000080: 6B 69 55 C7 2F 08 3B 94 2F 08 3B 94 2F 08 3B 94 kiU././././.;.
00000090: 69 59 E6 94 2C 08 3B 94 69 59 DB 94 3C 08 3B 94 iY...;iY.<.;.
000000A0: 69 59 DA 94 28 08 3B 94 F2 F7 F0 94 2D 08 3B 94 iY..(.;.....;.;.
000000B0: 2F 08 3A 94 13 08 3B 94 22 5A DA 94 2D 08 3B 94 /.:...;"Z..-.;.
000000C0: 22 5A E0 94 2E 08 3B 94 22 5A E5 94 2E 08 3B 94 "Z...;"Z....;.;.
000000D0: 52 69 63 68 2F 08 3B 94 00 00 00 00 00 00 00 00 Rich/.;.....
000000E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000F0: 50 45 00 00 4C 01 07 00 8A A4 96 5A 00 00 00 00 PE..L.....Z....
00000100: 00 00 00 00 E0 00 02 01 0B 01 0C 00 00 40 00 00 .....@.....
00000110: 00 3E 00 00 00 00 00 00 1D 11 01 00 00 10 00 00 .>.....
00000120: 00 10 00 00 00 00 40 00 00 10 00 00 00 02 00 00 .....@.....
00000130: 06 00 00 00 00 00 00 00 06 00 00 00 00 00 00 00 .....
00000140: 00 C0 01 00 00 04 00 00 00 00 00 00 03 00 40 81 .....@.
00000150: 00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00 .....
00000160: 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 .....
00000170: 74 91 01 00 3C 00 00 00 00 A0 01 00 3C 04 00 00 t...<.....<...
```



# 이진 파일 출력 예

```

00000000: 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 MZ.....
00000010: B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....@.....
00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 F0 00 00 00 .....
00000040: 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 .....!...L.!Th
00000050: 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F is program canno
00000060: 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 t be run in DOS
00000070: 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 mode....$.
00000080: 6B 69 55 C7 2F 08 3B 94 2F 08 3B 94 2F 08 3B 94 kiU././././.;.
00000090: 69 59 E6 94 2C 08 3B 94 69 59 DB 94 3C 08 3B 94 iY.,.;.iY..&lt..
000000A0: 69 59 DA 94 28 08 3B 94 F2 F7 F0 94 2D 08 3B 94 iY..(.;.....-.;
000000B0: 2F 08 3A 94 13 08 3B 94 22 5A DA 94 2D 08 3B 94 /.:...;."Z..-.;
000000C0: 22 5A E0 94 2E 08 3B 94 22 5A E5 94 2E 08 3B 94 "Z....;."Z....;
000000D0: 52 69 63 68 2F 08 3B 94 00 00 00 00 00 00 00 00 Rich/.;.....
000000E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000F0: 50 45 00 00 4C 01 07 00 8A A4 96 5A 00 00 00 00 PE..L.....Z....
00000100: 00 00 00 00 E0 00 02 01 0B 01 0C 00 00 40 00 00 .....@..
00000110: 00 3E 00 00 00 00 00 00 1D 11 01 00 00 10 00 00 .>.....
00000120: 00 10 00 00 00 00 40 00 00 10 00 00 00 02 00 00 .....@.....
00000130: 06 00 00 00 00 00 00 00 06 00 00 00 00 00 00 00 .....
00000140: 00 C0 01 00 00 04 00 00 00 00 00 00 03 00 40 81 .....@.
00000150: 00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00 .....
00000160: 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 .....
00000170: 74 91 01 00 3C 00 00 00 00 A0 01 00 3C 04 00 00 t...<.....<...

```

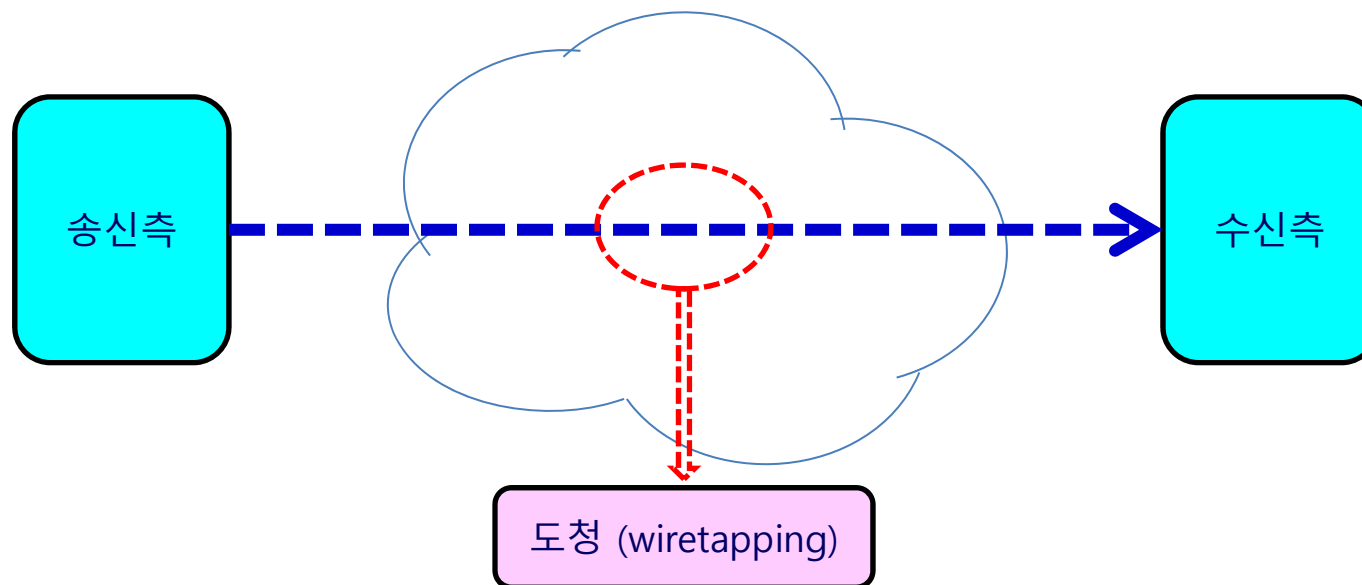


문자열을 16진수 문자로의 암호화

# 문자열 처리 응용 - 암호화 된 메시지 전송

## ◆ 송신측, 전송선로, 수신측

- 전송선로상에서 도청이 될 수 있고, 중요한 개인 정보 (신상정보, 은행계좌 및 비밀번호 등)가 유출될 수 있음
- 군사작전, 정부 비밀 정보 등에 대한 높은 보안 등급의 비밀 유지 필요
- 메시지 전송 전에 암호화 시키고, 암호화된 메시지를 전송하며, 메시지 수신 후 암호해독을 함으로써 개인정보 비밀 유지 가능





# 암호화된 메시지 전송

## ◆ 송신측

- 전송되는 메시지를 사전에 약속된 암호화 코드로 변환시켜 전송
- 전송 선로상에서 도청이 이루어져도, 암호화 코드를 모르는 경우, 원래의 메시지로 복원할 수 없거나, 복원시키기 위하여 매우 오랜 시간이 걸리게 함

## ◆ 수신측

- 수신된 데이터를 사전에 약속된 암호화 코드로 역변환시켜 원래의 메시지를 복원

## ◆ 사전에 약속된 암호화 코드, 암호화 방식

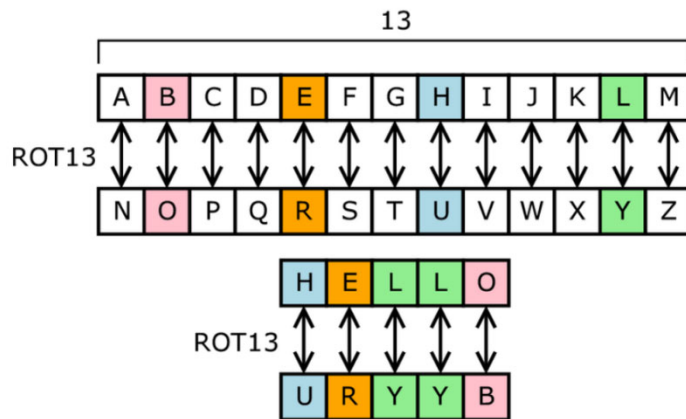
- 인터넷 은행 거래에서의 보안카드 번호 입력
- 인터넷 서버 접속 시 사용자 이름과 비밀번호 입력



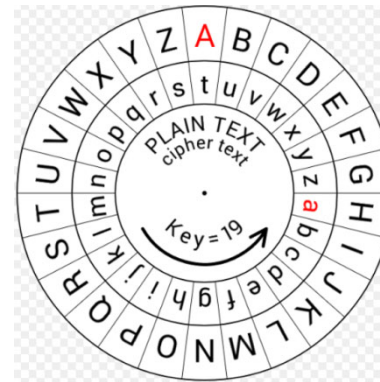
# 암호화 예 – Caesar Cipher (시저 암호)

## ◆ 시저 암호(Caesar cipher)

- 시저 암호는 암호학에서 다루는 간단한 치환암호의 일종
- 시저 암호는 약 기원전 100년경에 만들어져 로마의 장군인 시저 (Caesar)가 동맹군들과 소통하기 위해 만든 암호
- 암호화하고자 하는 내용을 알파벳 별로 일정한 거리만큼 밀어서 다른 알파벳으로 치환하는 방식이다. 예를 들어 13글자씩 밀어내는 시저 암호로 "HELLO"를 암호화하면 "URYJB"가 된다.



(a) 13자리씩 밀어 Caesar Cipher 암호화 하는 예



(b) Caesar Cipher 암호 해독 원판



# 간단한 암호화 개념을 사용하는 송신

## ◆ 송신 메시지의 문자열을 16진수 문자로 암호화

- ASCII code로 표현된 영문 한 글자 (character)의 8-비트 hexadecimal 값을 찾아 화면으로 16진수로 출력 (예: 0x12, 0xAF)
- 위에서 16진수의 상위 4비트 값과 하위 4비트 값을 각각 2개의 16진수 ASCII 문자 ('0' ~ '9', 'A' ~ 'F')로 출력  
예: 'T' (0x54) → 0x5, 0x4 (0 ~ 15의 값을 가짐)
- 0 ~ 15의 값을 가지는 정수를 16진수 ASCII 문자로 출력 ('0' ~ 'F')
- 위 기능을 사용하여, 한 줄의 ASCII 코드 메시지를 각 문자별 16진수의 상위 4비트와 하위 4비트 값을 구하고, 이에 대한 16진수 ASCII 문자로 출력 ('0' ~ 'F')  
예: "Test message" →  
**54657374206D657373616765206F6620**  
**323031373A30333A32350A**
- 참고: 0x54 : 'T', 0x65 : 'e', 0x73 : 's', 0x74 : 't'



# 간단한 암호화 개념을 사용하는 수신 및 복원

## ◆ 16진수 문자로 암호화된 메시지의 복원

- 16진수 문자로 암호화된 메시지로 부터 2문자 (실제로는 Hexadecimal 숫자) 단위로 읽고, 이에 대한 ASCII 코드를 찾아 변환

예: **54657374206D65737361676520  
6F6620323031373A30333A32350A**

→ **Test message**

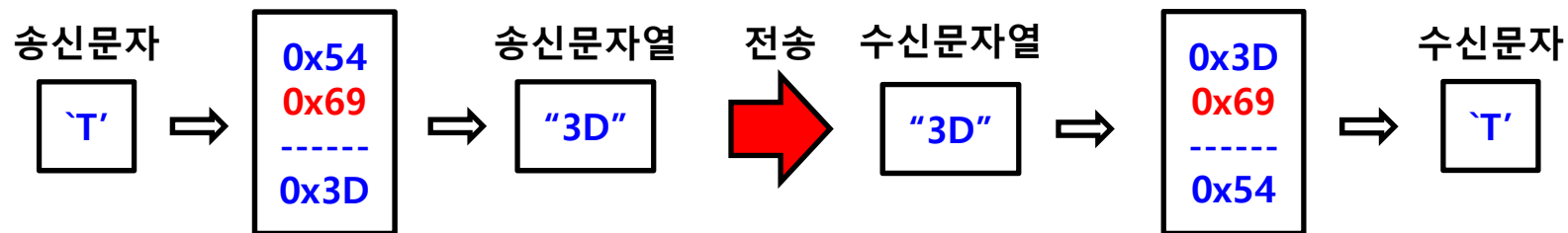
- 줄 바꿈 (0x0A )이 있는 경우, 이를 무시하고, 그 다음 16진수 문자를 2개 읽어 변환하여야 함



# 암호코드를 사용하는 16진수 문자로의 암호화 (1)

## ◆ 암호코드를 사용하는 방법

- 먼저 ASCII code로 표현된 영문자 (8-bit)를 주어진 **8-비트 암호코드**로 bit-wise exclusive OR를 실행한 후
- 암호화된 영문 한 글자 (character)의 8-비트 hexadecimal 값을 찾고, 이 16진수의 상위 4비트 값과 하위 4비트 값을 각각 2개의 16진수 ASCII 문자 ('0' ~ '9', 'A' ~ 'F')로 출력
- 수신단에서는 16진수 문자로 암호화된 메시지로 부터 2문자 (실제로는 Hexadecimal 숫자) 단위로 읽고, 이에 대한 ASCII 코드를 찾아 변환 ASCII 코드를 찾은 후, 이 ASCII 코드에 사전에 제공된 **8-비트 암호코드**로 bit-wise exclusive OR를 실행하여, 송신된 메시지를 복원



## 암호코드를 사용하는 16진수 문자로의 암호화 (3)

### ◆ 암호코드를 사용하여 전송 (cipher\_code = 0x69)

T: 0101 0100 (0x54)

cc: 0110 1001 (0x69)

xor: 0011 1101 (0x3D)

### ◆ 암호코드 **0x69**를 알고 있어야 해독 가능

### ◆ 만약, 128비트 이상의 암호코드를 사용하여, 암호화를 하는 경우, 이를 암호코드 없이 해독하여야 하는 경우 슈퍼 컴퓨터로도 수백년 이상 걸리게 됨



# cipherText(), deCipherText() 응용 프로그램

```
void test_simple_cipher_text()
{
    FILE *fp_msg, *fp_tx, *fp_rx, *fp_dump_msg, *fout;

    fp_msg = fopen("Message.txt", "r");
    if (fp_msg == NULL)
    {
        printf("Error in file open - Message.txt !!\n");
        exit(-1);
    }

    fp_tx = fopen("Transmit.txt", "w");
    if (fp_tx == NULL)
    {
        printf("Error in file open - Transmit.txt !!\n");
        exit(-1);
    }

    fout = fopen("Output.txt", "w");
    if (fout == NULL)
    {
        printf("Error in file open - Output.txt !!\n");
        exit(-1);
    }
}
```



```

printf("Generating cipher text with cipher-code (%#04x) ..\n", CIPHER_CODE);
cipherText(fp_msg, fp_tx, CIPHER_CODE);
fclose(fp_tx);

rewind(fp_msg);
fprintf(fout, "Binary dump of message.txt file: \n");
dumpBinaryFile(fp_msg, fout);

fp_tx = fopen("Transmit.txt", "r");
if (fp_tx == NULL)
{
    printf("Error in file open - Transmit.txt !!\n");
    exit(-1);
}

fprintf(fout, "\n===== \n");
fprintf(fout, "Binary dump of ciphered document: \n");
dumpBinaryFile(fp_tx, fout);
rewind(fp_msg);

printf("Generating de-ciphered text with cipher-code (%#04x) ..\n", CIPHER_CODE);
fprintf(fout, "\n===== \n");
fprintf(fout, "Generating de-ciphered text with cipher-code (%#04x) ..\n",
    CIPHER_CODE);
deCipherText(fp_tx, fout, CIPHER_CODE);

fclose(fp_msg);
fclose(fp_tx);
fclose(fout);

```

```

}

```





## ◆ Message.txt

Test message of 2020:05:20

The grand campaign will begin 2020:06:01 06:30AM, at Yeungnam Univ..

The second line of message.

The third line of message.

## ◆ Transmit.txt (cipher\_code = 0x00)

```
54657374206D65737373616765206F6620323032303A30353A32300A
```

```
546865206772616E642063616D706169676E2077696C6C20626567696E20323032303A30363A30312030363A3330414D2C206174205965756E676E616D20556E69762E2E0A
```

```
546865207365636F6E64206C696E65206F66206D657373736167652E0A
```

```
546865207468697264206C696E65206F66206D6573736167652E0A
```

## ◆ 화면출력

```
Generating cipher text with cipher-code (0000) ..
( 1-th Input string, length 27): Test message of 2020:05:20
( 2-th Input string, length 69): The grand campaign will begin 2020:06:01 06:30AM, at Yeungnam Univ..
( 3-th Input string, length 28): The second line of message.
( 4-th Input string, length 27): The third line of message.
( 5-th Input string, length 1):
( 6-th Input string, length 1):
Generating de-ciphered text with cipher-code (0000) ..

Received and deciphered message :
Test message of 2020:05:20
The grand campaign will begin 2020:06:01 06:30AM, at Yeungnam Univ..
The second line of message.
The third line of message.
```



## ◆ Output.txt (Cypher-code = 0x00)

Binary dump of message.txt file:

```
00000000: 54 65 73 74 20 6D 65 73 73 61 67 65 20 6F 66 20 Test message of
00000010: 32 30 32 30 3A 30 35 3A 32 30 0A 54 68 65 20 67 2020:05:20.The g
00000020: 72 61 6E 64 20 63 61 6D 70 61 69 67 6E 20 77 69 rand campaign wi
00000030: 6C 6C 20 62 65 67 69 6E 20 32 30 32 30 3A 30 36 ll begin 2020:06
00000040: 3A 30 31 20 30 36 3A 33 30 41 4D 2C 20 61 74 20 :01 06:30AM, at
00000050: 59 65 75 6E 67 6E 61 6D 20 55 6E 69 76 2E 2E 0A Yeungnam Univ...
00000060: 54 68 65 20 73 65 63 6F 6E 64 20 6C 69 6E 65 20 The second line
00000070: 6F 66 20 6D 65 73 73 61 67 65 2E 0A 54 68 65 20 of message..The
00000080: 74 68 69 72 64 20 6C 69 6E 65 20 6F 66 20 6D 65 third line of me
00000090: 73 73 61 67 65 2E 0A 0A 0A ssage....
```

=====

Binary dump of ciphered document:

```
00000000: 35 34 36 35 37 33 37 34 32 30 36 44 36 35 37 33 54657374206D6573
00000010: 37 33 36 31 36 37 36 35 32 30 36 46 36 36 32 30 73616765206F6620
00000020: 33 32 33 30 33 32 33 30 33 41 33 30 33 35 33 41 323032303A30353A
00000030: 33 32 33 30 30 41 0A 35 34 36 38 36 35 32 30 36 32300A.546865206
00000040: 37 37 32 36 31 36 45 36 34 32 30 36 33 36 31 36 772616E642063616
00000050: 44 37 30 36 31 36 39 36 37 36 45 32 30 37 37 36 D706169676E20776
00000060: 39 36 43 36 43 32 30 36 32 36 35 36 37 36 39 36 96C6C20626567696
00000070: 45 32 30 33 32 33 30 33 32 33 30 33 41 33 30 33 E20323032303A303
00000080: 36 33 41 33 30 33 31 32 30 33 30 33 36 33 41 33 63A30312030363A3
00000090: 33 33 30 34 31 34 44 32 43 32 30 36 31 37 34 32 330414D2C2061742
000000A0: 30 35 39 36 35 37 35 36 45 36 37 36 45 36 31 36 05965756E676E616
000000B0: 44 32 30 35 35 36 45 36 39 37 36 32 45 32 45 30 D20556E69762E2E0
000000C0: 41 0A 35 34 36 38 36 35 32 30 37 33 36 35 36 33 A.54686520736563
000000D0: 36 46 36 45 36 34 32 30 36 43 36 39 36 45 36 35 6F6E64206C696E65
000000E0: 32 30 36 46 36 36 32 30 36 44 36 35 37 33 37 33 206F66206D657373
000000F0: 36 31 36 37 36 35 32 45 30 41 0A 35 34 36 38 36 6167652E0A.54686
0000100: 35 32 30 37 34 36 38 36 39 37 32 36 34 32 30 36 5207468697264206
0000110: 43 36 39 36 45 36 35 32 30 36 46 36 36 32 30 36 C696E65206F66206
0000120: 44 36 35 37 33 37 33 36 31 36 37 36 35 32 45 30 D6573736167652E0
0000130: 41 0A 30 41 0A 30 41 0A A.OA.OA.
```

=====

Generating de-ciphered text with cipher-code (0000) ..

Test message of 2020:05:20

The grand campaign will begin 2020:06:01 06:30AM, at Yeungnam Univ..

The second line of message.

The third line of message.



## ◆ Message.txt

Test message of 2020:05:20

The grand campaign will begin 2020:06:01 06:30AM, at Yeungnam Univ..

The second line of message.

The third line of message.

## ◆ Transmit.txt (cipher\_code = 0x69)

3D0C1A1D49040C1A1A080E0C49060F495B595B5953595C535B5963

3D010C490E1B08070D490A08041908000E07491E000505490B0C0E0007495B595B5953595F53595849595F535A5928244549081D49300C1C070E070804493C07001F474763

3D010C491A0C0A06070D490500070C49060F49040C1A1A080E0C4763

3D010C491D01001B0D490500070C49060F49040C1A1A080E0C4763

## ◆ 화면출력

```
Generating cipher text with cipher-code (0x69) ..
( 1-th Input string, length 27): Test message of 2020:05:20
( 2-th Input string, length 69): The grand campaign will begin 2020:06:01 06:30AM, at Yeungnam Univ..
( 3-th Input string, length 28): The second line of message.
( 4-th Input string, length 27): The third line of message.
( 5-th Input string, length 1):
( 6-th Input string, length 1):
Generating de-ciphered text with cipher-code (0x69) ..
```

Received and deciphered message :

Test message of 2020:05:20

The grand campaign will begin 2020:06:01 06:30AM, at Yeungnam Univ..

The second line of message.

The third line of message.



## ◆ Output.txt (Cypher-code = 0x69)

Binary dump of message.txt file:

```
00000000: 54 65 73 74 20 6D 65 73 73 61 67 65 20 6F 66 20 Test message of
00000010: 32 30 32 30 3A 30 35 3A 32 30 0A 54 68 65 20 67 2020:05:20.The g
00000020: 72 61 6E 64 20 63 61 6D 70 61 69 67 6E 20 77 69 rand campaign wi
00000030: 6C 6C 20 62 65 67 69 6E 20 32 30 32 30 3A 30 36 ll begin 2020:06
00000040: 3A 30 31 20 30 36 3A 33 30 41 4D 2C 20 61 74 20 :01 06:30AM, at
00000050: 59 65 75 6E 67 6E 61 6D 20 55 6E 69 76 2E 2E 0A Yeungnam Univ...
00000060: 54 68 65 20 73 65 63 6F 6E 64 20 6C 69 6E 65 20 The second line
00000070: 6F 66 20 6D 65 73 73 61 67 65 2E 0A 54 68 65 20 of message..The
00000080: 74 68 69 72 64 20 6C 69 6E 65 20 6F 66 20 6D 65 third line of me
00000090: 73 73 61 67 65 2E 0A 0A 0A ssage....
```

=====  
Binary dump of ciphered document:

```
00000000: 33 44 30 43 31 41 31 44 34 39 30 34 30 43 31 41 3D0C1A1D49040C1A
00000010: 31 41 30 38 30 45 30 43 34 39 30 36 30 46 34 39 1A080E0C49060F49
00000020: 35 42 35 39 35 42 35 39 35 33 35 39 35 43 35 33 5B595B5953595C53
00000030: 35 42 35 39 36 33 0A 33 44 30 31 30 43 34 39 30 5B5963.3D010C490
00000040: 45 31 42 30 38 30 37 30 44 34 39 30 41 30 38 30 E1B08070D490A080
00000050: 34 31 39 30 38 30 30 30 45 30 37 34 39 31 45 30 41908000E07491E0
00000060: 30 30 35 30 35 34 39 30 42 30 43 30 45 30 30 30 00505490B0C0E000
00000070: 37 34 39 35 42 35 39 35 42 35 39 35 33 35 39 35 7495B595B5953595
00000080: 46 35 33 35 39 35 38 34 39 35 39 35 46 35 33 35 F53595849595F535
00000090: 41 35 39 32 38 32 34 34 35 34 39 30 38 31 44 34 A5928244549081D4
000000A0: 39 33 30 30 43 31 43 30 37 30 45 30 37 30 38 30 9300C1C070E07080
000000B0: 34 34 39 33 43 30 37 30 30 31 46 34 37 34 37 36 4493C07001F47476
000000C0: 33 0A 33 44 30 31 30 43 34 39 31 41 30 43 30 41 3.3D010C491A0C0A
000000D0: 30 36 30 37 30 44 34 39 30 35 30 30 30 37 30 43 06070D490500070C
000000E0: 34 39 30 36 30 46 34 39 30 34 30 43 31 41 31 41 49060F49040C1A1A
000000F0: 30 38 30 45 30 43 34 37 36 33 0A 33 44 30 31 30 080E0C4763.3D010
0000100: 43 34 39 31 44 30 31 30 30 31 42 30 44 34 39 30 C491D01001B0D490
0000110: 35 30 30 30 37 30 43 34 39 30 36 30 46 34 39 30 500070C49060F490
0000120: 34 30 43 31 41 31 41 30 38 30 45 30 43 34 37 36 40C1A1A080E0C476
0000130: 33 0A 36 33 0A 36 33 0A 3.63.63.
```

=====  
Generating de-ciphered text with cipher-code (0x69) ..

Test message of 2020:05:20

The grand campaign will begin 2020:06:01 06:30AM, at Yeungnam Univ..

The second line of message..

The third line of message..



**실습 10    main()**

# main()

```
/* main() for String, RandFileAccess, BinFileDump, Cipher (1) */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "MyString.h"
#include "Student.h"
#include "HandlingBinaryFile.h"
#include "CipherMessage.h"

void test_SelectionSortWords();
void test_RandomFileAccess_Student();
void test_BinaryFileDump();
void test_simple_cipher_text();

void main()
{
    int menu;

    while (1)
    {
        printf(" 1. Test selection sort for words\n");
        printf(" 2. Test random access file for students records\n");
        printf(" 3. Test binary file dump\n");
        printf(" 4. Test cipher text\n");
        printf("-1. Quit\n");
```



```
/* main() for String, RandFileAccess, BinFileDump, Cipher (2) */
```

```
    printf("Input menu : ");  
    scanf("%d", &menu);  
    if (menu == -1)  
        break;  
    switch (menu)  
    {  
    case 1:  
        test_SelectionSortWords();  
        break;  
    case 2:  
        test_RandomFileAccess_Student();  
        break;  
    case 3:  
        test_BinaryFileDump();  
        break;  
    case 4:  
        test_simple_cipher_text();  
        break;  
    default:  
        break;  
    }  
}
```





# 문자열의 파일 입력, 정렬 및 출력

Sorted words:

access	accomplish	achieve	activate	administration	adopt	advanced	advice	agenda	aim
alignment	analysis	analyst	application	apply	appreciate	approach	appropriate	approval	architecture
arrangement	arrive	aspect	assignment	attendee	authorization	background	bang	beforehand	benefit
blond	breakdown	brief	broadcast	budget	calendar	capability	capacity	capital	career
catalog	cautious	centralize	chair	chairperson	challenge	champion	clarify	client	closely
code	collaboration	commercial	common	communicate	compatible	competitor	compile	completed	complicated
concentrate	concept	concerned	conduct	conference	confidence	configuration	confirmation	confuse	congratulation
conquer	consideration	consist	consultant	contact	contain	convincing	cooperate	coordination	corporate
costly	coworker	credit	critically	current	customer	customize	deadline	define	definitely
delegate	deliver	deliverable	demand	demand	deployment	designer	detail	develop	device
direct	direction	discuss	division	documentation	dominate	download	draft	duration	duty
dynamic	edit	effective	electronic	elevator	embarrassing	encourage	engaged	environment	equipment
equivalent	establish	estimate	evaluation	evolve	excellent	exception	excited	existing	expand
expansion	expectation	expenditure	expertise	explanation	facility	familiarity	feature	fluent	form
formal	fulfill	full-time	function	gather	generate	gently	government	grade	graduate
guidance	handle	handover	hierarchical	hobby	host	impact	implementation	impress	impression
improve	inform	informative	infrastructure	initiate	installation	instruction	instructor	integrate	interactive
interest	interface	interview	introduction	involve	kickoff	latest	limit	logic	mainland
maintain	mapping	member	mention	messy	methodology	migrate	milestone	mission	modify
module	motivate	neat	notice	objective	offer	operational	opportunity	optimistic	option
orientation	oriented	otherwise	outsource	overall	part	participant	participate	patient	perform
perhaps	period	permission	phase	pick	popular	portfolio	post	practice	presentation
prevent	private	procedure	process	productive	professional	progress	project	promise	proposal
provide	query	rapidly	react	reasonable	receive	reconsider	reduce	referee	refine
release	remind	remotely	reply	reputation	request	requirement	resource	respectively	responsibility
resume	rough	scale	scenario	schedule	sector	security	seek	semester	senior
sensitive	separate	shape	share	similar	smoothly	snapshot	solid	solution	somewhat
sophisticated	specialty	specific	specification	stage	standard	storage	stream	subject	submit
successful	sufficient	suggestion	summary	supervise	supervisor	support	target	technical	thought
trial	troubleshoot	understanding	up-to-date	upload	usage	user-friendly	validate	variable	various
vendor	verify	version	victory	vital	widely	wonder	wonderful		





# 학생 레코드의 랜덤 파일 입출력

```
Array of students at initialization :
Student [ID: 21711000, Kim, G-M , (1990, 10, 5), (tel: +82-053-0805-1234), GPA: 3.57]
Student [ID: 21611075, Yoon, S-M , (1990, 4, 5), (tel: +82-053-0811-1550), GPA: 4.37]
Student [ID: 21411015, Hwang, S-S, (1989, 1, 10), (tel: +82-053-0817-1005), GPA: 2.72]
Student [ID: 21611054, Lee, K-M , (1991, 5, 15), (tel: +82-010-9112-9876), GPA: 3.35]
Student [ID: 21611340, Hong, G-M , (1990, 2, 5), (tel: +82-055-0810-5678), GPA: 3.89]
Student [ID: 21712056, Jang, S-M , (1990, 3, 15), (tel: +82-010-9112-1600), GPA: 4.42]
Student [ID: 21411017, Park, S-S , (1989, 7, 10), (tel: +82-034-0817-1098), GPA: 4.12]
Student [ID: 21511053, Choi, Y-H , (1992, 9, 25), (tel: +82-053-0845-5764), GPA: 3.85]
Student [ID: 21411017, Shin, D-J , (1988, 10, 3), (tel: +82-031-0817-1038), GPA: 3.21]
Student [ID: 21511053, Kwak, S-B , (1994, 11, 15), (tel: +82-002-0897-8778), GPA: 4.45]

Storing sorted students by increasing order of student ID into SortedStudent.txt ...
Student [ID: 21411015, Hwang, S-S, (1989, 1, 10), (tel: +82-053-0817-1005), GPA: 2.72]
Student [ID: 21411017, Park, S-S , (1989, 7, 10), (tel: +82-034-0817-1098), GPA: 4.12]
Student [ID: 21411017, Shin, D-J , (1988, 10, 3), (tel: +82-031-0817-1038), GPA: 3.21]
Student [ID: 21511053, Choi, Y-H , (1992, 9, 25), (tel: +82-053-0845-5764), GPA: 3.85]
Student [ID: 21511053, Kwak, S-B , (1994, 11, 15), (tel: +82-002-0897-8778), GPA: 4.45]
Student [ID: 21611054, Lee, K-M , (1991, 5, 15), (tel: +82-010-9112-9876), GPA: 3.35]
Student [ID: 21611075, Yoon, S-M , (1990, 4, 5), (tel: +82-053-0811-1550), GPA: 4.37]
Student [ID: 21611340, Hong, G-M , (1990, 2, 5), (tel: +82-055-0810-5678), GPA: 3.89]
Student [ID: 21711000, Kim, G-M , (1990, 10, 5), (tel: +82-053-0805-1234), GPA: 3.57]
Student [ID: 21712056, Jang, S-M , (1990, 3, 15), (tel: +82-010-9112-1600), GPA: 4.42]

Random access to Sorted_students.txt file (Student record length: 89) ...
Current file_position : 810
Student ( 9): Student [ID: 21712056, Jang, S-M , (1990, 3, 15), (tel: +82-010-9112-1600), GPA: 4.42]
Current file_position : 720
Student ( 8): Student [ID: 21711000, Kim, G-M , (1990, 10, 5), (tel: +82-053-0805-1234), GPA: 3.57]
Current file_position : 630
Student ( 7): Student [ID: 21611340, Hong, G-M , (1990, 2, 5), (tel: +82-055-0810-5678), GPA: 3.89]
Current file_position : 540
Student ( 6): Student [ID: 21611075, Yoon, S-M , (1990, 4, 5), (tel: +82-053-0811-1550), GPA: 4.37]
Current file_position : 450
Student ( 5): Student [ID: 21611054, Lee, K-M , (1991, 5, 15), (tel: +82-010-9112-9876), GPA: 3.35]
Current file_position : 360
Student ( 4): Student [ID: 21511053, Kwak, S-B , (1994, 11, 15), (tel: +82-002-0897-8778), GPA: 4.45]
Current file_position : 270
Student ( 3): Student [ID: 21511053, Choi, Y-H , (1992, 9, 25), (tel: +82-053-0845-5764), GPA: 3.85]
Current file_position : 180
Student ( 2): Student [ID: 21411017, Shin, D-J , (1988, 10, 3), (tel: +82-031-0817-1038), GPA: 3.21]
Current file_position : 90
Student ( 1): Student [ID: 21411017, Park, S-S , (1989, 7, 10), (tel: +82-034-0817-1098), GPA: 4.12]
Current file_position : 0
Student ( 0): Student [ID: 21411015, Hwang, S-S, (1989, 1, 10), (tel: +82-053-0817-1005), GPA: 2.72]
```



# 이진파일 Dump

```
00000000: 53 74 75 64 65 6E 74 20 5B 49 44 3A 20 32 31 34 Student [ID: 214
00000010: 31 31 30 31 35 2C 20 48 77 61 6E 67 2C 20 53 2D 11015, Hwang, S-
00000020: 53 2C 20 28 31 39 38 39 2C 20 20 31 2C 20 31 30 S, (1989, 1, 10
00000030: 29 2C 20 28 74 65 6C 3A 20 2B 38 32 2D 30 35 33 ), (tel: +82-053
00000040: 2D 30 38 31 37 2D 31 30 30 35 29 2C 20 47 50 41 -0817-1005), GPA
00000050: 3A 20 20 32 2E 37 32 5D 0D 0A 53 74 75 64 65 6E : 2.72]..Studen
00000060: 74 20 5B 49 44 3A 20 32 31 34 31 31 30 31 37 2C t [ID: 21411017,
00000070: 20 50 61 72 6B 2C 20 53 2D 53 20 2C 20 28 31 39 Park, S-S , (19
00000080: 38 39 2C 20 20 37 2C 20 31 30 29 2C 20 28 74 65 89, 7, 10), (te
00000090: 6C 3A 20 2B 38 32 2D 30 33 34 2D 30 38 31 37 2D 1: +82-034-0817-
000000A0: 31 30 39 38 29 2C 20 47 50 41 3A 20 20 34 2E 31 1098), GPA: 4.1
000000B0: 32 5D 0D 0A 53 74 75 64 65 6E 74 20 5B 49 44 3A 2]..Student [ID:
000000C0: 20 32 31 34 31 31 30 31 37 2C 20 53 68 69 6E 2C 21411017, Shin,
000000D0: 20 44 2D 4A 20 2C 20 28 31 39 38 38 2C 20 31 30 D-J , (1988, 10
000000E0: 2C 20 20 33 29 2C 20 28 74 65 6C 3A 20 2B 38 32 , 3), (tel: +82
000000F0: 2D 30 33 31 2D 30 38 31 37 2D 31 30 33 38 29 2C -031-0817-1038),
00000100: 20 47 50 41 3A 20 20 33 2E 32 31 5D 0D 0A 53 74 GPA: 3.21]..St
00000110: 75 64 65 6E 74 20 5B 49 44 3A 20 32 31 35 31 31 udent [ID: 21511
00000120: 30 35 33 2C 20 43 68 6F 69 2C 20 59 2D 48 20 2C 053, Choi, Y-H ,
```



# 메시지 암호화 및 전송, 역암호화 (Cypher-code = 0x69)

Binary dump of message.txt file:

```
00000000: 54 65 73 74 20 6D 65 73 73 61 67 65 20 6F 66 20 Test message of
00000010: 32 30 32 30 3A 30 35 3A 32 30 0A 54 68 65 20 67 2020:05:20.The g
00000020: 72 61 6E 64 20 63 61 6D 70 61 69 67 6E 20 77 69 rand campaign wi
00000030: 6C 6C 20 62 65 67 69 6E 20 32 30 32 30 3A 30 36 ll begin 2020:06
00000040: 3A 30 31 20 30 36 3A 33 30 41 4D 2C 20 61 74 20 :01 06:30AM, at
00000050: 59 65 75 6E 67 6E 61 6D 20 55 6E 69 76 2E 2E 0A Yeungnam Univ...
00000060: 54 68 65 20 73 65 63 6F 6E 64 20 6C 69 6E 65 20 The second line
00000070: 6F 66 20 6D 65 73 73 61 67 65 2E 0A 54 68 65 20 of message..The
00000080: 74 68 69 72 64 20 6C 69 6E 65 20 6F 66 20 6D 65 third line of me
00000090: 73 73 61 67 65 2E 0A 0A 0A ssage....
```

=====

Binary dump of ciphered document:

```
00000000: 33 44 30 43 31 41 31 44 34 39 30 34 30 43 31 41 3D0C1A1D49040C1A
00000010: 31 41 30 38 30 45 30 43 34 39 30 36 30 46 34 39 1A080E0C49060F49
00000020: 35 42 35 39 35 42 35 39 35 33 35 39 35 43 35 33 5B595B5953595C53
00000030: 35 42 35 39 36 33 0A 33 44 30 31 30 43 34 39 30 5B5963.3D010C490
00000040: 45 31 42 30 38 30 37 30 44 34 39 30 41 30 38 30 E1B08070D490A080
00000050: 34 31 39 30 38 30 30 30 45 30 37 34 39 31 45 30 41908000E07491E0
00000060: 30 30 35 30 35 34 39 30 42 30 43 30 45 30 30 30 00505490B0C0E000
00000070: 37 34 39 35 42 35 39 35 42 35 39 35 33 35 39 35 7495B595B5953595
00000080: 46 35 33 35 39 35 38 34 39 35 39 35 46 35 33 35 F53595849595F535
00000090: 41 35 39 32 38 32 34 34 35 34 39 30 38 31 44 34 A5928244549081D4
000000A0: 39 33 30 30 43 31 43 30 37 30 45 30 37 30 38 30 9300C1C070E07080
000000B0: 34 34 39 33 43 30 37 30 30 31 46 34 37 34 37 36 4493C07001F47476
000000C0: 33 0A 33 44 30 31 30 43 34 39 31 41 30 43 30 41 3.3D010C491A0C0A
000000D0: 30 36 30 37 30 44 34 39 30 35 30 30 30 37 30 43 06070D490500070C
000000E0: 34 39 30 36 30 46 34 39 30 34 30 43 31 41 31 41 49060F49040C1A1A
000000F0: 30 38 30 45 30 43 34 37 36 33 0A 33 44 30 31 30 080E0C4763.3D010
00000100: 43 34 39 31 44 30 31 30 30 31 42 30 44 34 39 30 C491D01001B0D490
00000110: 35 30 30 30 37 30 43 34 39 30 36 30 46 34 39 30 500070C49060F490
00000120: 34 30 43 31 41 31 41 30 38 30 45 30 43 34 37 36 40C1A1A080E0C476
00000130: 33 0A 36 33 0A 36 33 0A 3.63.63.
```

=====

Generating de-ciphered text with cipher-code (0x69) ..

Test message of 2020:05:20

The grand campaign will begin 2020:06:01 06:30AM, at Yeungnam Univ..

The second line of message.

The third line of message.



# **Oral Test 10**

# Oral Test

Q10.1 C 프로그램에서 파일 입력 및 파일 출력에 사용되는 fscanf(), fprintf(), fgets(), fputs()에 대하여 예를 들어 설명하라. 각 함수에서 사용가능한 형식 (format) 지정자에 대하여 설명할 것.

Q10.2 ASCII 코드표에서 숫자 (0 ~ 9), 대문자 ('A' ~ 'Z'), 소문자 ('a' ~ 'z')의 값이 어떤 구간에 있는지에 대하여 설명하라.

Q10.3 함수 void generateCipherText(FILE \*fp\_msg, FILE \*fp\_tx, unsigned char cipher\_code)에서 암호화 된 문자열이 어떻게 생성되며, 이 문자열을 주어진 암호를 모르는 경우 해석할 수 없는 이유에 대하여 설명하라.

Q10.4 함수 void deCipherText(FILE \*fp\_tx, FILE \*fp\_out, unsigned char cipher\_code)를 사용하여 암호화된 문자열을 원래의 문자열로 변환시키는 기능에 대하여 상세하게 설명하라.

