

lec01

lec01main.c

```
#include <stdio.h>

/* 이 프로그램은 무진장 간단한 예제 프로그램입니다.
   C 프로그램 입문을 환영합니다.
   이 프로그램을 작성한 후에 <Ctrl>+<F5>를 누르면 컴파일 후 실행이 됩니다.
*/

void main()
{
    printf("Hello world !");
}
```

lec02

lec02sum1.c

```
/* sum1.c */
/* 이 프로그램은 1 부터 100 까지의 합을 구하는 프로그램이다. */
/* 강의 주제 :
   프로그램이란 무엇인가.
   정수형 변수란 무엇인가.
   for 문의 사용법.
*/

#include <stdio.h>

main()
{
    int i;
    int sum = 0;

    for(i = 1; i <= 100; i++) {
        sum = sum + i;
    }
    printf("sum = %d\n",sum);
}

/* 결과는 다음과 같이 나온다.
sum = 5050
*/
```

lec03

lec03sum2.c

```
/* sum2.c */
/* 강의 주제 :
   함수란 무엇인가.
   함수를 사용하는 이유.
   함수의 시그니처.
   #include 문의 용도
   if 문의 사용법. (더불어 제어문에 대한 간략한 소개)
*/

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

int sum(int n)
{
    int total = 0;
    int i;

    if (n < 0) {
        fprintf(stderr, "Please use positive number.\n");
        exit(-1);
    }
    for(i = 0; i <= n; i++) {
        total = total + i;
    }
    return total;
}

main()
{
    printf("sum(10) = %d\n", sum(10));
    printf("sum(100) = %d\n", sum(100));
}

/* 결과는 다음과 같이 나온다.
sum(10) = 55
sum(100) = 5050
*/
```

lec04

lec04sum3.c

```
/* sum3.c */
/* 강의 주제 :
   재귀 함수란 무엇인가.
   compile error(syntax error, warning error), link error
   forward declaration
*/

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

int sum(int n);

main()
{
    // print("sum(10) = %d\n",sum(10)); // link error 발생
    /* warning C4013: 'print' undefined; assuming extern returning int */
    /* error LNK2001: unresolved external symbol _print */
    /* fatal error LNK1120: 1 unresolved externals */
    printf("sum(10) = %d\n",sum(10));
    printf("sum(100) = %d\n",sum(100));
}

int sum(int n)
{
    if (n < 0) {
        fprintf(stderr,"Please use positive number.\n");
        exit(-1);
    }
    // if (m == 0) return 0; // syntax error 발생
    /* error C2065: 'm' : undeclared identifier */
    if (n == 0) return 0;
    if (n == 1) return 1;
    return n + sum(n-1);
}

/* 결과는 다음과 같이 나온다.
sum(10) = 55
sum(100) = 5050
*/
```

lec05

lec05sum4.c

```
/* sum4.c */
/* 강의 주제 :
   efficiency
*/

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

int sum(int n);

main()
{
    printf("sum(10) = %d\n",sum(10));
    printf("sum(100) = %d\n",sum(100));
}

int sum(int n)
{
    if (n < 0) {
        fprintf(stderr,"Please use positive number.\n");
        exit(-1);
    }
    return n*(n+1)/2;
}

/* 결과는 다음과 같이 나온다.
sum(10) = 55
sum(100) = 5050
*/
```

lec06

lec06main.c

```
/* main.c */
/* 강의 주제 :
   header file 사용법
   실행문과 비실행문 (preprocessor directives)
   link에 대한 이야기
   library에 대한 이야기
*/

#include <stdio.h>
#include "sum.h"

main()
{
    printf("sum(10) = %d\n",sum(10));
    printf("sum(100) = %d\n",sum(100));
}

/* 결과는 다음과 같이 나온다.
sum(10) = 55
sum(100) = 5050
*/
```

lec06sum.c

```
/* sum.c */

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

int sum(int n)
{
    if (n < 0) {
        fprintf(stderr,"Please use positive number.\n");
        exit(-1);
    }
    return n*(n+1)/2;
}
```

lec06sum.h

```
#ifndef _SUM_H
#define _SUM_H

extern int sum(int n);

#endif
```

lec07

lec07sum5.c

```
/* sum5.c */
/* 강의 주제 :
   formatting
*/

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

int sum(int n);

main()
{
    int i,j,k=1;

    for (i = 1; i <= 10; i++) {
        for (j = 1; j <= 10; j++)
            printf("%5d,",sum(k++));
        printf("\n");
    }
}

int sum(int n)
{
    if (n < 0) {
        fprintf(stderr,"Please use positive number.\n");
        exit(-1);
    }
    return n*(n+1)/2;
}

/* 결과는 다음과 같이 나온다.
   1,   3,   6,   10,  15,  21,  28,  36,  45,  55,
   66,  78,  91,  105, 120, 136, 153, 171, 190, 210,
   231, 253, 276, 300, 325, 351, 378, 406, 435, 465,
   496, 528, 561, 595, 630, 666, 703, 741, 780, 820,
   861, 903, 946, 990, 1035, 1081, 1128, 1176, 1225, 1275,
   1326, 1378, 1431, 1485, 1540, 1596, 1653, 1711, 1770, 1830,
   1891, 1953, 2016, 2080, 2145, 2211, 2278, 2346, 2415, 2485,
   2556, 2628, 2701, 2775, 2850, 2926, 3003, 3081, 3160, 3240,
   3321, 3403, 3486, 3570, 3655, 3741, 3828, 3916, 4005, 4095,
   4186, 4278, 4371, 4465, 4560, 4656, 4753, 4851, 4950, 5050,
*/
```

lec08

lec08sum6.c

```
/* sum6.c */
/* 강의 주제 :
   formatting
   character, string (문자 상수, 문자열 상수)
   int putc(int c, FILE *stream);
   int putchar(int c);
*/

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

int sum(int n);

main()
{
    int i,j,k=1;

    printf("< summation table from 1 to 100 >\n");
    for (i = 0; i < 10; i++) {
        putchar('+');
        // 위 문장은 옆에 문장과 동일함: putc('+',stdout);
        for (j = 0; j < 10; j++)
            printf("-----+");
        putchar('\n');
        putchar('|');
        for (j = 0; j < 10; j++) {
            printf("%5d|",sum(k++));
        }
        printf("\n");
    }
    putchar('+');
    for (j = 0; j < 10; j++) printf("-----+");
    printf("\n");
}

int sum(int n)
{
    if (n < 0) {
        fprintf(stderr, "Please use positive number.\n");
        exit(-1);
    }
    return n*(n+1)/2;
}

/* 결과는 다음과 같이 나온다.
< summation table from 1 to 100 >
+-----+-----+-----+-----+-----+
|  1|   3|   6|  10|  15|  21|  28|  36|  45|  55|
+-----+-----+-----+-----+-----+
| 66|  78|  91| 105| 120| 136| 153| 171| 190| 210|
+-----+-----+-----+-----+-----+
| 231| 253| 276| 300| 325| 351| 378| 406| 435| 465|
+-----+-----+-----+-----+-----+
| 496| 528| 561| 595| 630| 666| 703| 741| 780| 820|
+-----+-----+-----+-----+-----+
| 861| 903| 946| 990| 1035| 1081| 1128| 1176| 1225| 1275|
+-----+-----+-----+-----+-----+
```


1326 1378 1431 1485 1540 1596 1653 1711 1770 1830
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1891 1953 2016 2080 2145 2211 2278 2346 2415 2485
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2556 2628 2701 2775 2850 2926 3003 3081 3160 3240
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3321 3403 3486 3570 3655 3741 3828 3916 4005 4095
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4186 4278 4371 4465 4560 4656 4753 4851 4950 5050
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

*/

lec09

lec09main.c

```
/* main.c */
/* 강의 주제 :
   변수와 타입
   integer, floating number, pointer
   integer arithmetic
   formatting
*/

#include <stdio.h>
main()
{
    int a = 20;
    int b = 30;
/* PC에서는 float 크기가 int 크기와 같기 때문에 double을 쓰는것이 좋다.
   float c = 20.0;
   float d = 30.0;
*/
    double c = 20.0;
    double d = 30.0;
    int *ap;
    double *cp;

    printf("sizeof(int) = %d, sizeof(float) = %d, sizeof(double) = %d\n",
           sizeof(int),sizeof(float),sizeof(double));
    printf("%d %f\n",a/b,c/d);

    ap = &a; cp = &c;
/* syntax error의 예
   ap = a; cp = c;
   warning C4047: '=' : 'int *' differs in levels of indirection from 'int '
   error C2115: '=' : incompatible types
*/

    printf("%x %x\n",ap,cp);
    printf("%d %f\n",*ap,*cp);

    *ap = 10.3;
/* warning C4244: '=' : conversion from 'const double ' to 'int ', possible loss of data */
    *cp = 10.3;
    printf("%d %f\n",*ap,*cp);

    *ap = *ap + 30;
    *cp = *cp + 30.0;
    printf("%d %f\n",a,c);
}

/* 결과는 다음과 같이 나온다.
sizeof(int) = 4, sizeof(float) = 4, sizeof(double) = 8
0 0.666667
12ff7c 12ff70
20 20.000000
10 10.300000
40 40.300000
*/
```

lec10

lec10swap1.c

```
/* swap1.c */
/* 강의 주제 :
   swapping
*/

#include <stdio.h>
main()
{
    int a = 10;
    int b = 20;
    int tmp;

    printf("before swapping: a = %d b = %d\n",a,b);
    tmp = a;
    a = b;
    b = tmp;
/* 아래 방식으로 했을때...
    a = b;
    b = a;
*/
    printf("after swapping: a = %d b = %d\n",a,b);
}

/* 결과는 다음과 같이 나온다.
before swapping: a = 10 b = 20
after swapping: a = 20 b = 10
*/
```

lec11

lec11swap2.c

```
/* swap2.c */
/* 강의 주제 :
   swapping
   global variable, local variable
   scope
*/

#include <stdio.h>

int a = 10;
int b = 20;

void swap()
{
    int tmp;

    tmp = a;
    a = b;
    b = tmp;
}

main()
{
    printf("before swapping: a = %d b = %d\n",a,b);
    swap();
    printf("after swapping: a = %d b = %d\n",a,b);
}

/* 결과는 다음과 같이 나온다.
before swapping: a = 10 b = 20
after swapping: a = 20 b = 10
*/
```

lec12

lec12swap3.c

```
/* swap3.c */
/* 강의 주제 :
   swapping
   parameter passing: call by value
   formal parameter, actual parameter
*/

#include <stdio.h>

void swap(int a,int b)
{
    int tmp;

    tmp = a;
    a = b;
    b = tmp;
}

main()
{
    int a = 10;
    int b = 20;
    printf("before swapping: a = %d b = %d\n",a,b);
    swap(a,b);
    printf("after swapping: a = %d b = %d\n",a,b);
}

/* 결과는 다음과 같이 나온다.
before swapping: a = 10 b = 20
after swapping: a = 10 b = 20
*/
```

lec13

lec13swap4.c

```
/* swap4.c */
/* 강의 주제 :
   swapping
   parameter passing: call by reference
   formal parameter, actual parameter
*/

#include <stdio.h>

void swap(int *a,int *b)
{
    int tmp;

    tmp = *a;
    *a = *b;
    *b = tmp;
}

main()
{
    int a = 10;
    int b = 20;
    printf("before swapping: a = %d b = %d\n",a,b);
    swap(&a,&b);
    printf("after swapping: a = %d b = %d\n",a,b);
}

/* 결과는 다음과 같이 나온다.
before swapping: a = 10 b = 20
after swapping: a = 20 b = 10
*/
```

lec14

lec14set1.c

```
/* set1.c */
/* 강의 주제 :
   initialization
   parameter passing: call by value
*/

#include <stdio.h>

void set_10(int a,int b)
{
    a = 10;
    b = 10;
}

main()
{
    int x;
    int y;
    /* warning 에러:
warning C4700: local variable 'x' used without having been initialized
warning C4700: local variable 'y' used without having been initialized
*/
    printf("before initialization: x = %d y = %d\n",x,y);
    set_10(x,y);
    printf("after initialization: x = %d y = %d\n",x,y);
}

/* 결과는 다음과 같이 나온다.
before initialization: x = -858993460 y = -858993460
after initialization: x = -858993460 y = -858993460
*/
```

lec15

lec15set2.c

```
/* set2.c */
/* 강의 주제 :
   initialization
   parameter passing: call by value
*/

#include <stdio.h>

void set_10(int *a,int *b)
{
    *a = 10;
    *b = 10;
}

main()
{
    int x;
    int y;
/* warning 에러:
warning C4700: local variable 'x' used without having been initialized
warning C4700: local variable 'y' used without having been initialized
*/
    printf("before initialization: x = %d y = %d\n",x,y);
    set_10(&x,&y);
    printf("after initialization: x = %d y = %d\n",x,y);
}

/* 결과는 다음과 같이 나온다.
before initialization: x = -858993460 y = -858993460
after initialization: x = 10 y = 10
*/
```


lec16

lec16main.c

```
/* main.c */
/* 강의 주제 :
   표준 입력 scanf() : call by reference의 전형적인 예
   while() 문
   무한루프
*/

#include <stdio.h>
#define TRUE (1)

main()
{
    while(TRUE) {
        int data;
        scanf("%d",&data);
        if (data < 0) break;
        printf("You typed %d\n",data);
    }
    printf("Good bye !\n");
}

/* 결과는 다음과 같이 나온다.
12
You typed 12
23
You typed 23
-2
Good bye !
*/
```

lec17

lec17main.c

```
/* main.c */
/* 강의 주제 :
   일차원 배열
   포인터와 배열의 관계
*/

#include <stdio.h>

void print1(int p[10])
{
    int i;
    for(i = 0; i < 10; i++) {
        printf("%d ", p[i]);
    }
    printf("\n");
}

void print2(int p[], int n)
{
    int i;
    for(i = 0; i < n; i++) {
        printf("%d ", p[i]);
    }
    printf("\n");
}

void print3(int *p, int n)
{
    int i;
    for(i = 0; i < n; i++) {
        printf("%d ", p[i]);
    }
    printf("\n");
}

void print4(int *p, int n)
{
    int i;
    for(i = 0; i < n; i++) {
        printf("%d ", *p);
        p++;
    }
    printf("\n");
}

void print5(int *p, int n)
{
    int i;
    for(i = 0; i < n; i++) {
        printf("%d ", *(p+i));
    }
    printf("\n");
}

void sub_10_from_data(int x[])
{
    int n;
```

```

    for(n = 0; n < 10; n++) {
        x[n] = x[n] - 10;
    }
}

void add_20_to_data(int data[],int max)
{
    int n;
    for(n = 0; n < max; n++) {
        data[n] = data[n] + 20;
    }
}

void add_data(int data[],int max,int how_much)
{
    int i;
    for(i = 0; i < max; i++) {
        data[i] = data[i] + how_much;
    }
}

main()
{
    int data[10];
    int i;

    for(i = 0; i < 10; i++) {
        data[i] = i*10;
    }
    print1(data);
    sub_10_from_data(data);
    print1(&data[0]);
    print2(data,10);
    print3(data,10);
    print4(data,10);
    print5(data,10);
    add_20_to_data(data,10);
    print2(data,10);
    add_data(data,10,30);
    print2(data,10);
}

```

```

/* 결과는 다음과 같이 나온다.
0 10 20 30 40 50 60 70 80 90
-10 0 10 20 30 40 50 60 70 80
-10 0 10 20 30 40 50 60 70 80
-10 0 10 20 30 40 50 60 70 80
-10 0 10 20 30 40 50 60 70 80
-10 0 10 20 30 40 50 60 70 80
-10 0 10 20 30 40 50 60 70 80
10 20 30 40 50 60 70 80 90 100
40 50 60 70 80 90 100 110 120 130
*/

```

lec18

lec18main.c

```
/* main.c */
/* 강의 주제 :
   이차원 배열 (행렬)
   이차원 배열과 일차원 배열과의 관계
   포배열과 포인터
*/

#include <stdio.h>

void print1(int p[3][4])
/*void print1(int p[][4])*/
/*void print1(int p[][])*/
{
    int i,j;
    for(i = 0; i < 3; i++) {
        for(j = 0; j < 4; j++)
            printf("%d ",p[i][j]);
        printf("\n");
    }
    printf("\n");
}

void print2(int p[],int row,int col)
{
    int i,j;
    for(i = 0; i < row; i++) {
        for(j = 0; j < col; j++)
            printf("%d ",p[i*col+j]);
        printf("\n");
    }
    printf("\n");
}

void print3(int *p,int row,int col)
{
    int i,j;
    for(i = 0; i < row; i++) {
        for(j = 0; j < col; j++)
            printf("%d ",p[i*col+j]);
        printf("\n");
    }
    printf("\n");
}

void print4(int *p,int row,int col)
{
    int i,j;
    for(i = 0; i < row; i++) {
        for(j = 0; j < col; j++) {
            printf("%d ",*p);
            p++;
        }
        printf("\n");
    }
    printf("\n");
}
```

```

void print5(int *p,int row,int col)
{
    int i,j;
    for(i = 0; i < row; i++) {
        for(j = 0; j < col; j++)
            printf("%d ",*(p+i*col+j));
        printf("\n");
    }
    printf("\n");
}

void sub_10_from_data(int x[],int row,int col)
{
    int n;
    for(n = 0; n < row*col; n++) {
        x[n] = x[n] - 10;
    }
}

void add_20_to_data(int *x,int row,int col)
{
    int n;
    for(n = 0; n < row*col; n++) {
        *(x+n) = *(x+n) + 20;
    }
}

void add_data(int data[],int row,int col,int how_much)
{
    int i,j;
    for(i = 0; i < row; i++)
        for(j = 0; j < col; j++)
            data[i*col+j] = data[i*col+j] + how_much;
}

main()
{
    int data[3][4];
    int i,j,k;

    k = 0;
    for(i = 0; i < 3; i++)
        for(j = 0; j < 4; j++)
            data[i][j] = 10*k++;

    for(i = 0; i < 3; i++) {
        for(j = 0; j < 4; j++)
            printf("%d ",data[i][j]);
        printf("\n");
    }
    printf("\n");

    print1(data);
    sub_10_from_data((int*)data,3,4);
    print2((int*)data,3,4);
    print3((int*)&data[0][0],3,4);
    print4((int*)&data[0][0],3,4);
    print5((int*)data,3,4);
    add_20_to_data((int*)data,3,4);
    print2((int*)data,3,4);
    add_data((int*)data,3,4,30);
    print2((int*)data,3,4);
}

```

/* 결과는 다음과 같이 나온다.

0 10 20 30
40 50 60 70
80 90 100 110

0 10 20 30
40 50 60 70
80 90 100 110

-10 0 10 20
30 40 50 60
70 80 90 100

-10 0 10 20
30 40 50 60
70 80 90 100

..... 이런식으로 나온다.

*/

lec19

lec19string1.c

```
/* string1.c */
/* 강의 주제 :
   string
   strlen() 함수
*/

#include <stdio.h>

/* version 1
int my_strlen(char *s)
{
int n;

for(n = 0; *s != '\0'; s++) {
n++;
}
return n;
}
*/

/* version 2
my_strlen(char *s)
{
char *p = s;

while(*p != '\0') {
p++;
}
return(p-s);
}
*/

// version 3
my_strlen(char *s)
{
char *p = s;

while(*p)
p++;
return(p-s);
}

main()
{
//char *p = "Kim";
//char p[] = {'K','i','m','\0'};
//char p[10] = {'K','i','m','\0'};
char p[10] = "Kim";

printf("length of \"kim\" is %d\n",my_strlen(p));
}

/* 결과는 다음과 같이 나온다.
length of "kim" is 3
*/
```

lec20

lec20string2.c

```
/* string2.c */
/* 강의 주제 :
   string
   strcpy() 함수
*/

#include <stdio.h>

/* version 1
void my_strcpy(char s[],char t[])
{
    int i;

    i = 0;
    while((s[i] = t[i]) != '\0')
        i++;
}
*/

/* version 2
void my_strcpy(char *s,char *t)
{
    while((*s = *t) != '\0') {
        s++;
        t++;
    }
}
*/

/* version 3
void my_strcpy(char *s,char *t)
{
    while((*s++ = *t++) != '\0');
}
*/

// version 4
void my_strcpy(char *s,char *t)
{
    while((*s++ = *t++));
}

main()
{
    /* runtime error
    char *x = "Kim";
    char *y;
    */
    /* runtime error
    char *x = "Kim";
    char *y = "Lee";
    */
    /* runtime error
    char *x = "Kim Taegyun";
    char y[4];
    */
    /*
```



```
char *x = "Kim";
char y[10] = "Lee";
*/
char *x = "Kim";
char y[10];
my_strcpy(y,x);
printf("%s %s\n",x,y);
}

/* 결과는 다음과 같이 나온다.
Kim Kim
*/
```

lec21

lec21string3.c

```
/* string3.c */
/* 강의 주제 :
   string
   strcmp() 함수
   ASCII 코드: 'a'는 97 'A'는 65
*/

#include <stdio.h>

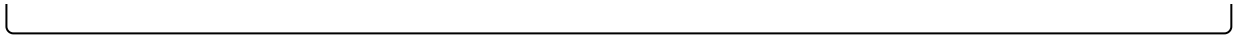
/* version 1
int my_strcmp(char s[],char t[])
{
    int i;

    i = 0;
    while (s[i] == t[i])
        if (s[i++] == '\0')
            return(0);
    return(s[i]-t[i]);
}
*/

// version 2
int my_strcmp(char *s,char *t)
{
    for( ; *s == *t; s++, t++)
        if (*s == '\0')
            return(0);
    return(*s - *t);
}

main()
{
    char *p = "Kwon";
    char *q = "Kim";
    int cmp = my_strcmp(p,q);
    if (cmp == 0) {
        printf("Kwon equals Kim.\n");
    } else if (cmp > 0) {
        printf("Kwon is greater than Kim.\n");
    } else if (cmp < 0) {
        printf("Kwon is less than Kim.\n");
    }
    /* 잘못된 프로그램
    if (p == q) {
        printf("Kwon equals Kim.\n");
    } else if (p > q) {
        printf("Kwon is greater than Kim.\n");
    } else if (p < q) {
        printf("Kwon is less than Kim.\n");
    }
    */
}

/* 결과는 다음과 같이 나온다.
Kwon is greater than Kim.
*/
```



lec22

lec22string4.c

```
/* string4.c */
/* 강의 주제 :
   string
   strcat() 함수
*/

#include <stdio.h>
#include <string.h>

void my_strcat(char s[],char t[])
{
    int i,j;

    i = j = 0;
    while(s[i] != '\0')
        i++;
    while((s[i++] = t[j++]) != '\0')
        ;
}

main()
{
    /* runtime error
    char *p = "Taegyun";
    char *q = "Kim";
    */
    /*
    char *p = "Taegyun";
    char q[20] = {'K','i','m','\0'};
    */
    char *p = "Taegyun";
    char q[20];

    strcpy(q, "Kim");
    my_strcat(q,p);
    printf("%s %s\n",p,q);
}

/* 결과는 다음과 같이 나온다.
Taegyun KimTaegyun
*/
```

lec23

lec23string5.c

```
/* string5.c */
/* 강의 주제 :
   string
   strdup() 함수
   dynamic allocation
   garbage
   lifetime
*/

#include <stdio.h>
#include <string.h>
#include <malloc.h>

char *my_strdup(char *s)
{
    char *p;

    p = (char *)malloc(strlen(s)+1);
    strcpy(p,s);
    return p;
}

main()
{
    char *x;

    // error case: strcpy(x,"Hello, my name is kim.");
    x = my_strdup("Hello, my name is kim.");
    printf("%s\n",x);
    free(x);
}

/* 결과는 다음과 같이 나온다.
Hello, my name is kim.
*/
```

lec24

lec24string6.c

```
/* string6.c */
/* 강의 주제 :
   string의 배열 (automatic allocation)
   string 함수 사용법과 주의 사항
*/

#include <stdio.h>
#include <string.h>
#define MAX (5)

void printName(char *p)
{
    printf("name = %s\n",p);
}

main()
{
    char names[5][10];
    int i;

    strcpy(names[0], "kim");
    strcpy(names[1], "lee");
    strcpy(names[2], "kwon");
    strcpy(names[3], "seong");
    strcpy(names[4], "park");

    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");

    names[1][2] = 'X';
    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");

    /* 이상한 현상 발생 ... 이유는 ...
    names[1][3] = 'Y';
    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");
    */

    strcpy(names[2], "abcdefghi");
    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");

    strcpy(names[2], "abcdefghij");
    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");
}

/* 결과는 다음과 같이 나온다.
name = kim
name = lee
name = kwon
name = seong
name = park

name = kim
name = leX
```

```
name = kwon  
name = seong  
name = park
```

```
name = kim  
name = leX  
name = abcdefghi  
name = seong  
name = park
```

```
name = kim  
name = leX  
name = abcdefghij  
name =  
name = park
```

```
*/
```

lec25

lec25string7.c

```
/* string7.c */
/* 강의 주제 :
   string의 배열 (static allocation)
   string 함수 사용법과 주의 사항
   이러한 경우는 데이터를 주로 read 할 때 만 사용함
*/

#include <stdio.h>
#include <string.h>
#define MAX (5)

void printName(char *p)
{
    printf("name = %s\n",p);
}

main()
{
    char *names[5];
    int i;

    names[0] = "kim";
    names[1] = "lee";
    names[2] = "kwon";
    names[3] = "seong";
    names[4] = "park";

    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");

    /* runtime error
    names[1][2] = 'X';
    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");
    */

    /* runtime error
    strcpy(names[2],"abcdefghi");
    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");
    */
}

/* 결과는 다음과 같이 나온다.
name = kim
name = lee
name = kwon
name = seong
name = park
*/
```


lec26

lec26string8.c

```
/* string8.c */
/* 강의 주제 :
   string 함수 사용법과 주의 사항
   이 프로그램은 컴파일 및 실행이 되지 않는다.
*/

#include <stdio.h>
#include <string.h>
#define MAX (5)

void printName(char *p)
{
    printf("name = %s\n",p);
}

main()
{
    char names[5][10];
    int i;

    /* syntax error :
       error C2106: '=' : left operand must be l-value
    */
    names[0] = "kim";
    names[1] = "lee";
    names[2] = "kwon";
    names[3] = "seong";
    names[4] = "park";

    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");
}
```

lec27

lec27string9.c

```
/* string9.c */
/* 강의 주제 :
   string의 배열 (포인터의 배열)
   string 함수 사용법과 주의 사항
   이러한 경우가 일반적인 사용법이다.
*/

#include <stdio.h>
#include <string.h>
#include <malloc.h>
#define MAX (5)

void printName(char *p)
{
    printf("name = %s\n",p);
}

main()
{
    char *names[5];
    int i;

    names[0] = strdup("kim");
    names[1] = strdup("lee");
    names[2] = strdup("kwon");
    names[3] = strdup("seong");
    names[4] = strdup("park");

    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");

    names[1][2] = 'X';
    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");

    /* 이상한 현상 발생 ... 이유는 ...
    names[1][3] = 'Y';
    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");
    */

    /* 이상한 현상 발생 혹은 runtime error ... 이유는 ...
    strcpy(names[2], "abcdefghijklmnopqrstuvwxy");
    */
    free(names[2]);
    names[2] = strdup("abcdefghijklmnopqrstuvwxy");
    for(i = 0; i < MAX; i++) printName(names[i]);
    printf("\n");
}

/* 결과는 다음과 같이 나온다.
name = kim
name = lee
name = kwon
name = seong
name = park
```

```
name = kim  
name = leX  
name = kwon  
name = seong  
name = park
```

```
name = kim  
name = lee  
name = abcdefghijklmnopqrstuvwxyz  
name = seong  
name = park
```

```
*/
```

lec28

lec28main.c

```
/* main.c */
/* 강의 주제 :
   comman line argument
*/

#include <stdio.h>
#include <string.h>

main(int argc, char *argv[])
{
    int i;

    /*
    printf("argc = %d\n",argc);
    for(i = 0; i < argc; i++) {
        printf("argv[%d] = %s\n",i,argv[i]);
    }
    */

    printf("You typed %d words in command line.\n",argc);
    for (i = 0; i < argc; i++) {
        printf("%s\n",argv[i]);
    }
}

/* 만약 도스창에서 실행 파일의 이름을 I.exe로 바꾼 후
"I am Taegyun Kim" 이라고 치면 결과는 다음과 같이 나온다.
You typed 4 words in command line.
I
am
Taegyun
Kim
*/
```

lec29

lec29calendar.c

```
/* calendar.c */
/* 강의 주제 :
   간단한 응용 프로그램의 예
   달력 프로그램
   switch 문
*/

#define YES (1)
#define NO (0)
#include <stdio.h>
#include <stdlib.h>

void printHeading(int year,int month)
{
    printf("          %d/%d\n",month,year);
    printf(" Sun Mon Tue Wed Thu Fri Sat\n");
}

char isLeapYear(int year)
{
    if (year % 400 == 0) return YES;
    if (year % 100 == 0) return NO;
    if (year % 4 == 0) return YES;
    return NO;
    /*
    if (year%4 == 0 && year%100 != 0 || year%400 == 0) return YES;
    else return NO;
    */
}

int getDaysToYear(int year)
{
    int totalDays = 0;
    int i;

    for(i = 1; i <= year; i++) {
        if (isLeapYear(i) == YES)
            totalDays = totalDays+366;
        else
            totalDays = totalDays+365;
    }
    return totalDays;
}

int getDaysToMonth(char leapFlag,int month)
{
    int daysOfMonth[2][12] = {
        {31,28,31,30,31,30,31,31,30,31,30,31},
        {31,29,31,30,31,30,31,31,30,31,30,31} };
    int totalDays = 0;
    int i;

    for(i = 0; i < month; i++) {
        if (leapFlag == YES)
            totalDays = totalDays+daysOfMonth[1][i];
        else
            totalDays = totalDays+daysOfMonth[0][i];
    }
}
```

```

    }
    return totalDays;
}

int getTotalDays(int year,int month,char flag)
{
    int daysBeforeThisYear;
    int daysBeforeThisMonth;
    int totalDays;

    daysBeforeThisYear = getDaysToYear(year-1);
    daysBeforeThisMonth = getDaysToMonth(flag,month-1);
    totalDays = daysBeforeThisYear + daysBeforeThisMonth;
    return totalDays;
}

void fillDays(int d[],int start,int last)
{
    int i;
    int day=1;

    for(i = start; day <= last; i++) {
        d[i] = day;
        day++;
    }
}

void printCalendar(int year,int month)
{
    int day[6][7];
    int totalDays;
    char leapYearFlag;
    int startDate;
    int lastDate;
    int i,j;

    for(i = 0; i < 6; i++)
        for(j = 0; j < 7; j++)
            day[i][j] = 0;
    printHeading(year,month);
    leapYearFlag = isLeapYear(year);
    totalDays = getTotalDays(year,month,leapYearFlag) + 1 /* today */;
    startDate = totalDays%7;
    switch(month) {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12: lastDate = 31;
                break;
        case 2:
            if(leapYearFlag) {
                lastDate = 29;
            } else {
                lastDate = 28;
            }
            break;
        default:
            lastDate = 30;
    }
    fillDays((int*)&day[0][0],startDate,lastDate);
    for(i = 0; i < 6; i++) {
        for(j = 0; j < 7; j++) {

```

```

        if (day[i][j] != 0)
            printf("%4d",day[i][j]);
        else
            printf("   ");
    }
    printf("\n");
}
}

main(int argc,char *argv[])
{
    int year;
    int month;

    if (argc != 3) {
        fprintf(stderr,"Error: Wrong number of argument.\n\
Usage: calendar <month> <year>\n");
        exit(-1);
    }
    month = atoi(argv[1]);
    year = atoi(argv[2]);
    if (year < 1) {
        fprintf(stderr,"Error: Invalid year.\n\
Usage: calendar <month> <year>\n");
        exit(-1);
    }
    if (month < 1 || month > 12) {
        fprintf(stderr,"Error: Invalid month.\n\
Usage: calendar <month> <year>\n");
        exit(-1);
    }
    printCalendar(year,month);
}

```

/* 만약 도스창에서 실행 파일의 이름을 cal.exe로 바꾼 후
"calendar 10 2002" 이라고 치면 결과는 다음과 같이 나온다.

```

10/2002
Sun Mon Tue Wed Thu Fri Sat
          1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

```

*/

lec30

lec30io1.c

```
/* io1.c */
/* 강의 주제 :
   같은 정수 값을 입력해도 프로그램 내부에서 처리하는 방법은 다름
*/

#include <stdio.h>
#define MAX (10)
main()
{
    int x;
    float y;
    char buffer[MAX];

    printf("Type a integer number : ");
    scanf("%d",&x);
    printf("You typed : %d\n",x);

    printf("Type a integer number : ");
    scanf("%f",&y);
    printf("You typed : %f\n",y);

    printf("Type a integer number : ");
    scanf("%s",buffer);
    printf("You typed : %s\n",buffer);
}

/* 결과는 다음과 같이 나온다.
Type a integer number : 15
You typed : 15
Type a integer number : 15
You typed : 15.000000
Type a integer number : 15
You typed : 15
*/
```


lec31

lec31io2.c

```
/* io2.c */
/* 강의 주제 :
   정상적인 사용 방식
*/

#include <stdio.h>
#define MAX (10)
main()
{
    int x;
    float y;
    double z;
    char buffer[MAX];

    printf("Type a integer number : ");
    scanf("%d",&x);
    printf("integer : %d\n",x);

    printf("Type a float number : ");
    scanf("%f",&y);
    printf("float : %f\n",y);

    printf("Type a double number : ");
    scanf("%lf",&z);
    printf("double : %lf\n",z);

    printf("Type a string : ");
    scanf("%s",buffer);
    printf("string : %s\n",buffer);
}

/* 결과는 다음과 같이 나온다.
Type a integer number : 15
integer : 15
Type a float number : 32.1
float : 32.099998
Type a double number : 32.1
double : 32.100000
Type a string : kim
string : kim
*/
```

lec32

lec32io3.c

```
/* io3.c */
/* 강의 주제 :
   character 단위의 입력 처리
   atoi, atof
*/

#include <stdio.h>
#define MAX (100)

int getline(char s[],int limit)
{
    int c,i;

    i = 0;
    while(--limit > 0 && (c=getchar()) != EOF && c != '\n')
        s[i++] = c;
    s[i] = '\0';
    return(i);
}

int atoi(char s[])
{
    int i, n, sign;

    for(i = 0; s[i] == ' ' || s[i] == '\n' || s[i] == '\t'; i++);
    sign = 1;
    if (s[i] == '+' || s[i] == '-')
        sign = (s[i++]=='+') ? 1 : -1;
    for(n = 0; s[i] >= '0' && s[i] <= '9'; i++)
        n = 10 * n + s[i] - '0';
    return(sign * n);
}

double atof(char s[])
{
    double val,power;
    int i,sign;

    for(i = 0; s[i] == ' ' || s[i] == '\n' || s[i] == '\t'; i++);
    sign = 1;
    if (s[i] == '+' || s[i] == '-')
        sign = (s[i++]=='+') ? 1 : -1;
    for(val = 0; s[i] >= '0' && s[i] <= '9'; i++)
        val = 10 * val + s[i] - '0';
    if (s[i] == '.') i++;
    for(power = 1; s[i] >= '0' && s[i] <= '9'; i++) {
        val = 10 * val + s[i] - '0';
        power *= 10;
    }
    return(sign * val / power);
}

main()
{
    int x;
    double y;
    char buffer[MAX];
```

```
printf("Type a integer number : ");
getline(buffer,MAX);
x = atoi(buffer);
printf("integer : %d\n",x);

printf("Type a double number : ");
getline(buffer,MAX);
y = atof(buffer);
printf("double : %f\n",y);

printf("Type a string : ");
getline(buffer,MAX);
printf("string : %s\n",buffer);
}
```

/* 결과는 다음과 같이 나온다.

```
Type a integer number : 15
integer : 15
Type a double number : 32.1
double : 32.100000
Type a string : kim
string : kim
*/
```

lec33

lec33io4.c

```
/* io3.c */
/* 강의 주제 :
   파일 입출력
*/

#include <stdio.h>
#include <stdlib.h>
#define MAX (100)

int getline(char s[],int limit,FILE *infile)
{
    int c,i;

    i = 0;
    while(--limit > 0 && (c=getc(infile)) != EOF && c != '\n')
        s[i++] = c;
    s[i] = '\0';
    return(i);
}

main(int argc,char *argv[])
{
    FILE *infile;
    FILE *outfile;
    int x;
    float y;
    char buffer[MAX];

    if (argc != 3) {
        fprintf(stderr,"Error: Wrong number of argument.\n\
Usage: test <input> <output>\n");
        exit(-1);
    }
    infile = fopen(argv[1],"r");
    outfile = fopen(argv[2],"w");
    fscanf(infile,"%d\n",&x);
    fprintf(outfile,"integer read : %d\n",x);
    fscanf(infile,"%f\n",&y);
    fprintf(outfile,"float read : %f\n",y);
    getline(buffer,MAX,infile);
    fprintf(outfile,"string read : %s\n",buffer);
    fclose(infile);
    fclose(outfile);
}

/* 결과는 다음과 같이 나온다.
수업 시간에 설명 ...
*/
```

lec34

lec34sort1.c

```
/* sort1.c */
/* 강의 주제 :
   sorting (bubble sort)
*/

#include <stdio.h>
#define MAX    (100)

void sort(int v[],int n)
{
    int i,j;
    int tmp;

    for(i = 0; i < n-1; i++) {
        for(j = 0; j < n-i-1; j++) {
            if (v[j] > v[j+1]) {
                tmp = v[j];
                v[j] = v[j+1];
                v[j+1] = tmp;
            }
        }
    }
}

main()
{
    int data[MAX];
    int i,ndata;

    printf("Type the number of data: ");
    scanf("%d",&ndata);
    if (ndata > MAX) {
        printf("Too many data\n");
        exit(-1);
    }
    for(i = 0; i < ndata; i++) {
        printf("Type the %dth data: ",i+1);
        scanf("%d",&data[i]);
    }
    sort(data,ndata);
    printf("-- sorted data --\n");
    for(i = 0; i < ndata; i++) {
        printf("%d\n",data[i]);
    }
}

/* 결과는 다음과 같이 나온다.
Type the number of data: 5
Type the 1th data: 7
Type the 2th data: 3
Type the 3th data: 8
Type the 4th data: 1
Type the 5th data: 2
-- sorted data --
1
2
3
```

7
8
*/

lec35

lec35sort2.c

```
/* sort2.c */
/* 강의 주제 :
   sorting (character 단위 입력)
   end of file 이용
*/

#include <stdio.h>
#include <stdlib.h>
#define LINES    (100)
#define MAXINT   (10)

int getline(char s[],int limit)
{
    int c,i;

    i = 0;
    while(--limit > 0 && (c=getchar()) != EOF && c != '\n')
        s[i++] = c;
    if (c == '\n')
        s[i++] = c;
    s[i] = '\0';
    return(i);
}

int readlines(int data[], int maxlines)
{
    int len,nlines;
    char line[MAXINT];

    nlines = 0;
    while((len = getline(line,MAXINT)) > 0)
        if (nlines >= maxlines)
            return(-1);
        else {
            line[len-1] = '\0';
            data[nlines++] = atoi(line);
        }
    return(nlines);
}

void sort(int v[],int n)
{
    int i,j;
    int tmp;

    for(i = 0; i < n-1; i++) {
        for(j = 0; j < n-i-1; j++) {
            if (v[j] > v[j+1]) {
                tmp = v[j];
                v[j] = v[j+1];
                v[j+1] = tmp;
            }
        }
    }
}

main()
```

```

{
    int data[LINES];
    int i,nlines;

    if ((nlines = readlines(data,LINES)) >= 0) {
        sort(data,nlines);
        printf("-- sorted data --\n");
        for (i = 0; i < nlines; i++) {
            printf("%d\n",data[i]);
        }
    } else {
        printf("input too big to sort\n");
    }
}

/* 결과는 다음과 같이 나온다. (파일 이용 가능: sort < data.txt)
7
3
8
1
2
^Z
-- sorted data --
1
2
3
7
8
*/

```


lec36

lec36sort3.c

```
/* sort3.c */
/* 강의 주제 :
   sorting (문자 열 데이터)
   end of file 이용
*/

#include <stdio.h>
#include <string.h>
#include <malloc.h>

#define LINES    (100)
#define MAXLEN   1000

int getline(char s[],int limit)
{
    int c,i;

    i = 0;
    while(--limit > 0 && (c=getchar()) != EOF && c != '\n')
        s[i++] = c;
    if (c == '\n')
        s[i++] = c;
    s[i] = '\0';
    return(i);
}

int readlines(char *lineptr[], int maxlines)
{
    int len,nlines;
    char *p, line[MAXLEN];

    nlines = 0;
    while((len = getline(line,MAXLEN)) > 0)
        if (nlines >= maxlines)
            return(-1);
        else if ((p = (char*)malloc(len)) == NULL)
            return(-1);
        else {
            line[len-1] = '\0';
            strcpy(p,line);
            lineptr[nlines++] = p;
        }
    return(nlines);
}

void writelines(char *lineptr[],int nlines)
{
    int i;

    for(i = 0; i < nlines; i++)
        printf("%s\n",lineptr[i]);
}

void sort(char *v[],int n)
{
    int i,j;
    char *tmp;
```

```

for(i = 0; i < n-1; i++) {
    for(j = 0; j < n-i-1; j++) {
        if (strcmp(v[j],v[j+1]) > 0) {
            tmp = v[j];
            v[j] = v[j+1];
            v[j+1] = tmp;
        }
    }
}

main()
{
    char *lineptr[LINES];
    int nlines;

    if ((nlines = readlines(lineptr,LINES)) >= 0) {
        sort(lineptr,nlines);
        printf("-- sorted data --\n");
        writelines(lineptr,nlines);
    } else {
        printf("input too big to sort\n");
    }
}

/* 결과는 다음과 같이 나온다. (파일 이용 가능: sort < data.txt)
kim taegyun
seong sunyong
kim suehwan
choi jinho
jo jangwu
^Z
-- sorted data --
choi jinho
jo jangwu
kim suehwan
kim taegyun
seong sunyong
*/

```

lec37

lec37search1.c

```
/* search1.c */
/* 강의 주제 :
   linear search
*/

#include <stdio.h>
#define MAX (10)
#define TRUE (1)

int search(int what,int v[],int n)
{
    int i;

    for (i = 0; i < n; i++)
        if (what == v[i])
            return i;
    return -1;
}

main()
{
    int data[MAX] = { 31, 19, 26, 52, 1, 98, 90, 22, 9, 12 };
    int dataToFind;
    int foundIndex;

    while (TRUE) {
        printf("Type number to find: ");
        scanf("%d",&dataToFind);
        if (dataToFind < 0) break;
        foundIndex = search(dataToFind,data,MAX);
        if (foundIndex >= 0) {
            printf("%d is found at position %d.\n",dataToFind,foundIndex);
        } else { /* foundIndex == -1 */
            printf("%d is not found.\n",dataToFind);
        }
    }
}

/* 결과는 다음과 같이 나온다.
Type number to find: 31
31 is found at position 0.
Type number to find: 98
98 is found at position 5.
Type number to find: 17
17 is not found.
Type number to find: 12
12 is found at position 9.
Type number to find: -1
*/
```

lec38

lec38search2.c

```
/* search2.c */
/* 강의 주제 :
   binary search
*/
#include <stdio.h>
#define MAX (10)
#define TRUE (1)

int search(int what,int v[],int n)
{
    int low, high, mid;
    low = 0;
    high = n - 1;
    while(low <= high) {
        mid = (low+high) / 2;
        if (what < v[mid])
            high = mid - 1;
        else if (what > v[mid])
            low = mid + 1;
        else /* found match */
            return mid;
    }
    return -1;
}

main()
{
    int data[MAX] = { 1, 9, 12, 19, 22, 26, 31, 52, 90, 98 };
    int dataToFind;
    int foundIndex;

    while (TRUE) {
        printf("Type number to find: ");
        scanf("%d",&dataToFind);
        if (dataToFind < 0) break;
        foundIndex = search(dataToFind,data,MAX);
        if (foundIndex >= 0) {
            printf("%d is found at position %d.\n",dataToFind,foundIndex);
        } else { /* foundIndex == -1 */
            printf("%d is not found.\n",dataToFind);
        }
    }
}

/* 결과는 다음과 같이 나온다.
Type number to find: 1
1 is found at position 0.
Type number to find: 26
26 is found at position 5.
Type number to find: 17
17 is not found.

Type number to find: 98
98 is found at position 9.
Type number to find: -1
*/
```

lec39

lec39struct1.c

```
/* struct1.c */
/* 강의 주제 :
   structure (record)
*/

#include <stdio.h>

struct point {
    int x;
    int y;
};

void findCenterPoint(struct point *p, struct point q, struct point r)
{
    p->x = (q.x + r.x) / 2;
    p->y = (q.y + r.y) / 2;
}

void printPoint(char *name, struct point p)
{
    printf("%s = (%d,%d)\n", name, p.x, p.y);
}

main()
{
    struct point a;
    struct point b;
    struct point c;

    a.x = 10;
    a.y = 10;

    b.x = 20;
    b.y = 30;

    findCenterPoint(&c, a, b);
    printPoint("a", a);
    printPoint("b", b);
    printPoint("c", c);
}

/* 결과는 다음과 같이 나온다.
a = (10,10)
b = (20,30)
c = (15,20)
*/
```

lec40

lec40struct2.c

```
/* struct2.c */
/* 강의 주제 :
   structure (record)
   initialization
*/

#include <stdio.h>

struct point {
    int x;
    int y;
};

void findCenterPoint(struct point *p, struct point q, struct point r)
{
    p->x = (q.x + r.x) / 2;
    p->y = (q.y + r.y) / 2;
}

void printPoint(char *name, struct point p)
{
    printf("%s = (%d,%d)\n", name, p.x, p.y);
}

main()
{
    struct point a = { 10, 10 };
    struct point b = { 20, 30 };
    struct point c;

    findCenterPoint(&c, a, b);
    printPoint("a", a);
    printPoint("b", b);
    printPoint("c", c);
}

/* 결과는 다음과 같이 나온다.
a = (10,10)
b = (20,30)
c = (15,20)
*/
```

lec41

lec41struct3.c

```
/* struct3.c */
/* 강의 주제 :
   structure (record)
   전형적인 사용법
   call by reference 의 효용성
*/

#include <stdio.h>
#include <malloc.h>

typedef struct point {
    int x;
    int y;
} Point, *PointPtr;

/* very different from
struct point {
    int x;
    int y;
} Point;
*/

void findCenterPoint(Point *p, Point *q, Point *r)
{
    p->x = (q->x + r->x) / 2;
    p->y = (q->y + r->y) / 2;
}

void printPoint(char *name, PointPtr p)
{
    printf("%s = (%d,%d)\n", name, p->x, p->y);
}

main()
{
    Point *pa = (Point *)malloc(sizeof(Point));
    Point *pb = (Point *)malloc(sizeof(Point));
    Point *pc = (Point *)malloc(sizeof(Point));

    pa->x = 10;
    pa->y = 10;

    pb->x = 20;
    pb->y = 30;

    findCenterPoint(pc, pa, pb);
    printPoint("a", pa);
    printPoint("b", pb);
    printPoint("c", pc);

    free(pa);
    free(pb);
    free(pc);
}

/* 결과는 다음과 같이 나온다.
a = (10,10)
```

```
b = (20,30)
c = (15,20)
*/
```


lec42

lec42list1.c

```
/* list1.c */
/* 강의 주제 :
   linked list
   structure
   dynamic allocation
*/

#include <stdio.h>
#include <malloc.h>

typedef struct _node {
    int data;
    struct _node *next;
} node,*nodePtr;

void insertAtHead(nodePtr *list_pointer,int d)
{
    nodePtr newNode;

    if (*list_pointer == NULL) {
        newNode = (nodePtr)malloc(sizeof(node));
        newNode->data = d;
        newNode->next = NULL;
    } else {
        newNode = (nodePtr)malloc(sizeof(node));
        newNode->data = d;
        newNode->next = *list_pointer;
    }
    *list_pointer = newNode;
}

void insertAtTail(nodePtr *list_pointer,int d)
{
    nodePtr newNode;
    nodePtr tmp = *list_pointer;

    newNode = (nodePtr)malloc(sizeof(node));
    newNode->data = d;
    newNode->next = NULL;
    if (*list_pointer == NULL) {
        *list_pointer = newNode;
    } else {
        while(tmp->next != NULL) {
            tmp = tmp->next;
        }
        tmp->next = newNode;
    }
}

void print(nodePtr list)
{
    printf("< ");
    while (list != NULL) {
        printf("%d ",list->data);
        list = list->next;
    }
    printf(">\n");
}
```

```

}

int nOfList(nodePtr list)
{
    int n = 0;
    while(list != NULL) {
        n++;
        list = list->next;
    }
    return n;
}

void freeAll(nodePtr list)
{
    nodePtr tmp;
    while(list != NULL) {
        tmp = list;
        list = list->next;
        free(tmp);
    }
}

main()
{
    nodePtr list = NULL;

    insertAtHead(&list,10);
    insertAtHead(&list,20);
    insertAtHead(&list,30);
    insertAtHead(&list,40);
    insertAtTail(&list,50);
    insertAtTail(&list,60);
    insertAtTail(&list,70);
    insertAtTail(&list,80);
    print(list);
    printf("number of list = %d\n",nOfList(list));
    freeAll(list);
}

/* 결과는 다음과 같이 나온다.
< 40 30 20 10 50 60 70 80 >
number of list = 8
*/

```

lec43

lec43list2.c

```
/* 강의 주제 :
   linked list
   global variable 이용
*/

#include <stdio.h>
#include <malloc.h>

typedef struct _node {
    int data;
    struct _node *next;
} node,*nodePtr;

nodePtr list = NULL;

void insertAtHead(int d)
{
    nodePtr newNode;

    if (list == NULL) {
        newNode = (nodePtr)malloc(sizeof(node));
        newNode->data = d;
        newNode->next = NULL;
    } else {
        newNode = (nodePtr)malloc(sizeof(node));
        newNode->data = d;
        newNode->next = list;
    }
    list = newNode;
}

void insertAtTail(int d)
{
    nodePtr newNode;
    nodePtr tmp = list;

    newNode = (nodePtr)malloc(sizeof(node));
    newNode->data = d;
    newNode->next = NULL;
    if (list == NULL) {
        list = newNode;
    } else {
        while(tmp->next != NULL) {
            tmp = tmp->next;
        }
        tmp->next = newNode;
    }
}

void print()
{
    nodePtr tmp = list;

    printf("< ");
    while (tmp != NULL) {
        printf("%d ",tmp->data);
        tmp = tmp->next;
    }
}
```

```

    }
    printf(">\n");
}

int nOfList()
{
    nodePtr tmp = list;

    int n = 0;
    while(tmp != NULL) {
        n++;
        tmp = tmp->next;
    }
    return n;
}

void freeAll()
{
    nodePtr tmp;
    while(list != NULL) {
        tmp = list;
        list = list->next;
        free(tmp);
    }
}

main()
{
    insertAtHead(10);
    insertAtHead(20);
    insertAtHead(30);
    insertAtHead(40);
    insertAtTail(50);
    insertAtTail(60);
    insertAtTail(70);
    insertAtTail(80);
    print();
    printf("number of list = %d\n",nOfList());
    freeAll();
}

/* 결과는 다음과 같이 나온다.
< 40 30 20 10 50 60 70 80 >
number of list = 8
*/

```

lec44

lec44sin1.c

```
/* sin1.c */
/* 강의 주제 :
   텍스트 모드에서 sin() 그래프 그리기: 세로 방향
*/

#include <stdio.h>
#include <math.h>

#define YMAX    (40)
#define XMAX    (20)

#define M_PI    (3.141592)

void drawSine(FILE *fp);

main()
{
    FILE *fp = fopen("graph.txt","w");
    FILE *fp = stdout;
    fprintf(fp,"Sine Graph\n");
    drawSine(fp);
    fprintf(fp,"OK\n");
    fclose(fp);
}

void drawSine(FILE *fp)
{
    int i;
    int j;
    double step = 2*M_PI/XMAX;
    double x = 0;

    for(i = 0; i < YMAX/2; i++) fprintf(fp,"-");
    fprintf(fp,"*");
    for(i = 0; i < YMAX/2; i++) fprintf(fp,"-");
    fprintf(fp,"\n");

    for(i = 0; i < XMAX; i++) {
        char lineBuffer[YMAX+1];
        double y;
        int yPrime;

        x = x + step;
        y = sin(x);

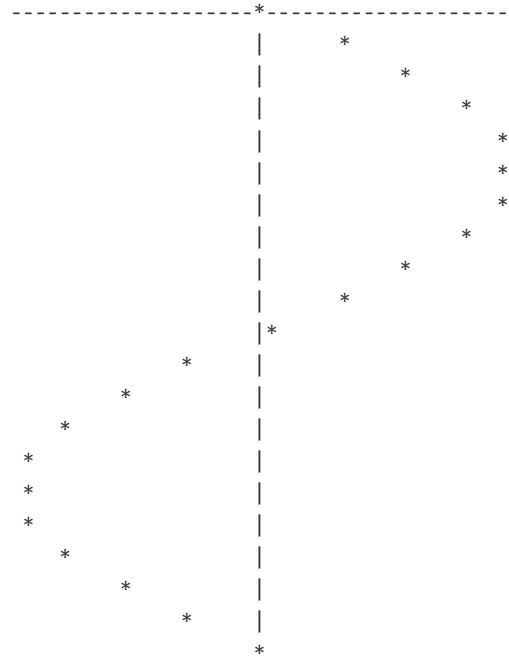
        yPrime = (int)((YMAX/2)*y + (YMAX/2+1));

        for(j = 0; j < YMAX+1; j++) lineBuffer[j] = ' ';
        lineBuffer[YMAX/2] = '|';
        lineBuffer[yPrime] = '*';

        for(j = 0; j < YMAX+1; j++) fprintf(fp,"%c",lineBuffer[j]);
        fprintf(fp,"\n");
    }
}

/* 결과는 다음과 같이 나온다.
```

Sine Graph



OK
*/

lec45

lec45sin2.c

```
/* sin2.c */
/* 강의 주제 :
   텍스트 모드에서 sin() 그래프 그리기: 가로 방향
*/

#include <stdio.h>
#include <math.h>

#define XMAX    (60)
#define YMAX    (20)

#define M_PI    (3.141592)

void drawSine(FILE *fp);

main()
{
    FILE *fp = fopen("graph.txt","w");
    FILE *fp = stdout;
    fprintf(fp,"Sine graph\n");
    drawSine(fp);
    fprintf(fp,"OK\n");
    fclose(fp);
}

void drawSine(FILE *fp)
{
    char screenBuffer[YMAX+1][XMAX+1];
    int i,j;
    double step = 2*M_PI/XMAX;
    double x = 0;

    for(i = 0; i < YMAX+1; i++)
        for(j = 0; j < XMAX+1; j++)
            screenBuffer[i][j] = ' ';

    for(i = 0; i < YMAX+1; i++) screenBuffer[i][0] = '|';
    screenBuffer[YMAX/2+1][0] = '*';
    for(i = 1; i < XMAX+1; i++) screenBuffer[YMAX/2+1][i] = '-';

    for(i = 1; i < XMAX+1; i++) {
        double y;
        int yPrime;

        x = x + step;
        y = sin(x);

        yPrime = (int)(-(YMAX/2)*y + (YMAX/2+1));
        screenBuffer[yPrime][i] = '*';
    }

    for(i = 0; i < YMAX+1; i++) {
        for(j = 0; j < XMAX+1; j++) {
            fprintf(fp,"%c",screenBuffer[i][j]);
        }
        fprintf(fp,"\n");
    }
}
```


lec46

lec46StdAfx.cpp

```
// stdafx.cpp : source file that includes just the standard includes
//   lec44.pch will be the pre-compiled header
//   stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"

// TODO: reference any additional headers you need in STDAFX.H
// and not in this file
```

lec46lec44.cpp

```
// lec44.cpp : Defines the entry point for the application.
//

#include "stdafx.h"
#include "resource.h"
#include <math.h>

#define MAX_LOADSTRING 100

/* manually added by taegyun kim */
#define XORIGIN (10)
#define YORIGIN (10)
#define HEIGHT (200)
#define WIDTH (500)
#define M_PI (3.141592)
/*****/

// Global Variables:
HINSTANCE hInst; // current instance
TCHAR szTitle[MAX_LOADSTRING]; // The title bar text
TCHAR szWindowClass[MAX_LOADSTRING]; // The title bar text

// Forward declarations of functions included in this code module:
ATOM MyRegisterClass(HINSTANCE hInstance);
BOOL InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK About(HWND, UINT, WPARAM, LPARAM);

void DrawSine(HDC hdc)
{
    double step = 2*M_PI/WIDTH;
    double x = 0;
    int oldX = XORIGIN;
    int oldY = YORIGIN;

    MoveToEx(hdc, XORIGIN, YORIGIN, NULL);
    LineTo(hdc, XORIGIN, YORIGIN+2*HEIGHT);

    MoveToEx(hdc, XORIGIN, YORIGIN+HEIGHT, NULL);
    LineTo(hdc, XORIGIN+WIDTH, YORIGIN+HEIGHT);

    for (int nY = XORIGIN; nY <= XORIGIN+WIDTH; nY++) {
```

```

for (int px = XORIGIN, px += XORIGIN+WIDTH, pY++) {
    double y = sin(x);
    int yPrime = (int)(-HEIGHT * y + (YORIGIN+HEIGHT));

    MoveToEx(hdc,oldX,oldY,NULL);
    LineTo(hdc,px,yPrime);

//     SetPixel(hdc,px,yPrime,RGB(0,0,0));

    oldX = px;
    oldY = yPrime;
    x = x + step;
}
}

int APIENTRY WinMain(HINSTANCE hInstance,
                    HINSTANCE hPrevInstance,
                    LPSTR lpCmdLine,
                    int nCmdShow)
{
    // TODO: Place code here.
    MSG msg;
    HACCEL hAccelTable;

    // Initialize global strings
    LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadString(hInstance, IDC_LEC44, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    // Perform application initialization:
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }

    hAccelTable = LoadAccelerators(hInstance, (LPCTSTR)IDC_LEC44);

    // Main message loop:
    while (GetMessage(&msg, NULL, 0, 0))
    {
        if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }

    return msg.wParam;
}

//
// FUNCTION: MyRegisterClass()
//
// PURPOSE: Registers the window class.
//
// COMMENTS:
//
// This function and its usage is only necessary if you want this code
// to be compatible with Win32 systems prior to the 'RegisterClassEx'
// function that was added to Windows 95. It is important to call this function
// so that the application will get 'well formed' small icons associated
// with it.
//

```

```

//
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEX wcex;

    wcex.cbSize = sizeof(WNDCLASSEX);

    wcex.style          = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc    = (WNDPROC)WndProc;
    wcex.cbClsExtra     = 0;
    wcex.cbWndExtra     = 0;
    wcex.hInstance     = hInstance;
    wcex.hIcon          = LoadIcon(hInstance, (LPCTSTR)IDI_LEC44);
    wcex.hCursor        = LoadCursor(NULL, IDC_ARROW);
    wcex.hbrBackground  = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszMenuName   = (LPCSTR)IDC_LEC44;
    wcex.lpszClassName = szWindowClass;
    wcex.hIconSm        = LoadIcon(wcex.hInstance, (LPCTSTR)IDI_SMALL);

    return RegisterClassEx(&wcex);
}

//
// FUNCTION: InitInstance(HANDLE, int)
//
// PURPOSE: Saves instance handle and creates main window
//
// COMMENTS:
//
//     In this function, we save the instance handle in a global variable and
//     create and display the main program window.
//
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    HWND hWnd;

    hInst = hInstance; // Store instance handle in our global variable

    hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);

    if (!hWnd)
    {
        return FALSE;
    }

    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    return TRUE;
}

//
// FUNCTION: WndProc(HWND, unsigned, WORD, LONG)
//
// PURPOSE: Processes messages for the main window.
//
// WM_COMMAND   - process the application menu
// WM_PAINT     - Paint the main window
// WM_DESTROY   - post a quit message and return
//
//
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    int wmId, wmEvent;
    PAINTSTRUCT ps;

```

```

PAINTSTRUCT ps,
HDC hdc;

switch (message)
{
    case WM_COMMAND:
        wmId    = LOWORD(wParam);
        wmEvent = HIWORD(wParam);
        // Parse the menu selections:
        switch (wmId)
        {
            case IDM_ABOUT:
                DialogBox(hInst, (LPCTSTR)IDD_ABOUTBOX, hWnd, (DLGPROC)About);
                break;
            case IDM_EXIT:
                DestroyWindow(hWnd);
                break;
            default:
                return DefWindowProc(hWnd, message, wParam, lParam);
        }
        break;
    case WM_PAINT:
        hdc = BeginPaint(hWnd, &ps);
        // TODO: Add any drawing code here...
        RECT rt;
        GetClientRect(hWnd, &rt);
        DrawSine(hdc);
        EndPaint(hWnd, &ps);
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    default:
        return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

// Message handler for about box.
LRESULT CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_INITDIALOG:
            return TRUE;

        case WM_COMMAND:
            if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
            {
                EndDialog(hDlg, LOWORD(wParam));
                return TRUE;
            }
            break;
    }
    return FALSE;
}
}

```

lec46StdAfx.h

```

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently

```

```

//

#if !defined(AFX_STDAFX_H_A9DB83DB_A9FD_11D0_BFD1_444553540000__INCLUDED_)
#define AFX_STDAFX_H_A9DB83DB_A9FD_11D0_BFD1_444553540000__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#define WIN32_LEAN_AND_MEAN           // Exclude rarely-used stuff from Windows headers

// Windows Header Files:
#include <windows.h>

// C RunTime Header Files
#include <stdlib.h>
#include <malloc.h>
#include <memory.h>
#include <tchar.h>

// Local Header Files

// TODO: reference additional headers your program requires here

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX_STDAFX_H_A9DB83DB_A9FD_11D0_BFD1_444553540000__INCLUDED_)

```

lec46lec44.h

```

#if !defined(AFX_LEC44_H_9650CFBC_DEC6_489B_9B65_D58A0CBF18C9__INCLUDED_)
#define AFX_LEC44_H_9650CFBC_DEC6_489B_9B65_D58A0CBF18C9__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "resource.h"

#endif // !defined(AFX_LEC44_H_9650CFBC_DEC6_489B_9B65_D58A0CBF18C9__INCLUDED_)

```

lec46resource.h

```

//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by LEC44.RC
//
#define IDR_MAINFRAME                128
#define IDD_LEC44_DIALOG             102
#define IDD_ABOUTBOX                 103
#define IDS_APP_TITLE                103
#define IDM_ABOUT                    104
#define IDM_EXIT                     105
#define IDS_HELLO                    106
#define IDI_LEC44                    107

```

```
#define IDI_SMALL 108
#define IDC_LEC44 109
#define IDC_MYICON 2
#define IDC_STATIC -1
// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS

#define _APS_NEXT_RESOURCE_VALUE 129
#define _APS_NEXT_COMMAND_VALUE 32771
#define _APS_NEXT_CONTROL_VALUE 1000
#define _APS_NEXT_SYMED_VALUE 110
#endif
#endif
```