Chapter 12

저장 클래스, 연계, 메모리 관리

Section 1 저장 클래스
Section 2 변수 유효 범위
Section 3 메모리 영역 할당

무엇을 학습할까요?

- 변수 유효 범위
- auto 변수, static 변수
- malloc( ) , free( )
- 실전 프로젝트(연습문제 풀이, 총 9 문제)

기초 다지기

저장 클래스	수명	범위	선언 방법
auto	자동	블록	블록 { } 안에서
register	자동	블록	블록 { } 안에서 register 변수
static	정적	파일	함수 밖에서 static 변수

기초 다지기

- int *pt ;
- pt = (int *)malloc(max * sizeof(int);
- free(pt)

프로그래밍 연습 문제 p662

01 리스트 12.4의 프로그램을 다시 작성하라.
이번에는 전역 변수를 사용하지 말라.

프로그래밍 연습 소스 01

```c
#include <stdio.h>
void critic(int * u);
int main(void)
{
    int units; /* units now local , extern int units; */
    printf("How many pounds to a firkin of butter?\n");
    scanf("%d", &units);
    while (units != 56)
        critic(&units);
    printf("You must have looked it up!\n");
    return 0;
}
```

프로그래밍 연습 소스 01

```c
void critic(int * u)
{
    printf("No luck, my friend. Try again.\n");
    scanf("%d", u);
}
// or use a return value:
// units = critic();
// and have critic look like this:
/*
int critic(void)
{
    int u;
    printf("No luck, my friend. Try again.\n");
    scanf("%d", &u);
    return u;
}
```

```
How many pounds to a firkin of butter?
14
No luck, my friend. Try again.
56
You must have looked it up!
```

02

휘발유 소비를 미국에서는 갤런당 마일 수로,
유럽에서는 100킬로미터당 리터 수로 계산한다.

다음은 미국식 모드로 계산할 것인지 유럽식 모드로 계산할 것인지를
사용자가 선택하고, 데이터를 읽고,
연료 소비를 계산하는 프로그램의 일부이다.

02

```
// pe12-2b.c
// pel 12-2a.c와 함께 컴파일하라.
#include <stdio.h>
#include "pe12-2a.h"
int main(void)
{
        ......................
     scanf("%d", &mode);
     printf("종료.\n");
    return 0;
}
```

02

다음은 몇 번의 실행 예이다.

```
유럽식 모드는 0, 미국식 모드는 1을 입력하시오: 0
주행한 거리를 킬로미터 단위로 입력하시오: 600
소비한 휘발유의 양을 리터 단위로 입력하시오: 78.8
유럽식: 100킬로미터당 13.13 리터입니다.
유럽식 모드는 0, 미국식 모드는 1을 입력하시오(끝내려면 -1): 1
주행한 거리를 마일 단위로 입력하시오: 434
```

02

```
소비한 휘발유의 양을 갤런 단위로 입력하시오: 12.7
미국식: 갤런당 34.2 마일입니다.
유럽식 모드는 0, 미국식 모드는 1을 입력하시오(끝내려면 -1) : 3
모드 지정이 잘못되었습니다. 모드 1 (미국식)을 사용합니다.
주행한 거리를 마일 단위로 입력하시오: 388
소비한 휘발유의 양을 갤런 단위로 입력하시오: 15.3
미국식: 갤런당 25.4 마일입니다.
유럽식 모드는 0, 미국식 모드는 1을 입력하시오(끝내려면 -1) : -1
종료.
```

02

사용자가 모드를 잘못 지정하면, 잘못된 지정이라는 메시지를 출력한 후,
바로 이전의 모드를 사용한다.
프로그램이 동작할 수 있도록 pe12-2a.h 헤더 파일과
pe12-2a.c 소스 코드 파일을 작성하라.
pe12-2a.c 소스 코드 파일은 파일 범위,
내부 연계 변수를 세 개 정의해야 한다.
하나는 모드를 나타내고, 또 하나는 주행한 거리를 나타내고,
마지막 하나는 소비한 휘발유의 양을 나타낸다.
get_info() 함수는 모드 설정에 따라 적절한 데이터를 입력하도록 사용자를 안내하고, 사용자의 응답을 파일 범위 변수에 저장한다.
show_info() 함수는 모드 설정에 따라 연료 소비를 계산하여 출력한다.
사용자가 숫자 입력으로 반응한다고 가정한다.

프로그래밍 연습 소스 02

```c
// pe12-2a.h
#define METRIC 0
#define US 1
void set_mode(int m);
void get_info(void);
void show_info(void);
```

프로그래밍 연습 소스 02

```c
#include <stdio.h>
#include "pe12-2a.h"
static int mode = METRIC;
static double distance;
static double fuel;
const char modestr[2][10] = { "Metric", "US" };
void set_mode(int m)
{
    if (m == METRIC || m == US)
        mode = m;

    else
    {
        printf("Invalid mode specified. Mode %d", mode);
        printf("(%s) used.\n", modestr[mode]);
    }
}
```

프로그래밍 연습 소스 02

```c
void get_info(void)
{
    if (mode == METRIC)
        printf("Enter distance traveled in kilometers: ");
    else
        printf("Enter distance traveled in miles: ");

    scanf("%lf", &distance);
    if (mode == METRIC)
        printf("Enter fuel consumed in liters: ");
    else
        printf("Enter fuel consumed in gallons: ");
    scanf("%lf", &fuel);
}
```

프로그래밍 연습 소스 02

```c
void show_info(void)
{
    printf("Fuel consumption is ");
    if (mode == METRIC)
        printf("%.2f liters per 100 km.\n", 100 * fuel / distance);
    else
        printf("%.1f miles per gallon.\n", distance / fuel);
}
```

프로그래밍 연습 소스 02

```c
// pe12-2b.c
// compile with pe12-2a.c
#include <stdio.h>
#include "pe12-2a.h"
int main(void)
{
    int mode;
    printf("Enter 0 for metric mode, 1 for US mode: ");
    scanf("%d", &mode);
```

프로그래밍 연습 소스 02

```
10      while (mode >= 0)
11      {
12          set_mode(mode);
13          get_info();
14          show_info();
15          printf("Enter 0 for metric mode, 1 for US mode");
16          printf(" (-1 to quit): ");
17          scanf("%d", &mode);
18      }
19      printf("Done.\n");
20      return 0;
21  }
```

프로그래밍 연습 소스 02

```
Enter 0 for metric mode, 1 for US mode: 0
Enter distance traveled in kilometers: 600
Enter fuel consumed in liters: 78.8
Fuel consumption is 13.13 liters per 100 km.
Enter 0 for metric mode, 1 for US mode (-1 to quit): 13.13
Invalid mode specified. Mode 0(Metric) used.
Enter distance traveled in kilometers: Enter fuel consumed in liters: 1
Fuel consumption is 769.23 liters per 100 km.
Enter 0 for metric mode, 1 for US mode (-1 to quit): 434
Invalid mode specified. Mode 0(Metric) used.
Enter distance traveled in kilometers: 12.7
Enter fuel consumed in liters: 54.2
Fuel consumption is 426.77 liters per 100 km.
Enter 0 for metric mode, 1 for US mode (-1 to quit): 3
Invalid mode specified. Mode 0(Metric) used.
Enter distance traveled in kilometers: 388
Enter fuel consumed in liters: 15.3
Fuel consumption is 3.94 liters per 100 km.
Enter 0 for metric mode, 1 for US mode (-1 to quit): -1
Done.
```

프로그래밍 연습 문제 p663

03

프로그래밍 연습②의 프로그램을 다시 설계하라.
이번에는 자동 변수만 사용하라.
다시 설계된 프로그램도 동일한 사용자 인터페이스를 가져야 한다.
사용자가 모드를 선택하면,
선택한 모드에 따라 적절하게 데이터를 입력하도록 사용자를 안내하고,
연료 소비를 계산하여 출력해야 한다.
자동 변수를 사용해야 하므로 함수 호출들이 달라져야 할 것이다.

프로그래밍 연습 소스 03

```c
//pe12-3a.h
#define METRIC 0
#define US 1
#define USE_RECENT 2
void check_mode(int *pm);
void get_info(int mode, double * pd, double * pf);
void show_info(int mode, double distance, double fuel);
```

프로그래밍 연습 소스 03

```c
// pe12-3a.c
// compile with pe12-3b.c
#include <stdio.h>
#include "pe12-3a.h"
void check_mode(int *pm)
{
    if (*pm != METRIC && *pm != US)
    {
        printf("Invalid mode specified. Mode %d\n", *pm);
        printf("Previous mode will be used.\n");
        *pm = USE_RECENT;
    }
}
```

프로그래밍 연습 소스 03

```c
void get_info(int mode, double * pd, double * pf)
{
    if (mode == METRIC)
        printf("Enter distance traveled in kilometers: ");
    else
        printf("Enter distance traveled in miles: ");
    scanf("%lf", pd);
    if (mode == METRIC)
        printf("Enter fuel consumed in liters: ");
    else
        printf("Enter fuel consumed in gallons: ");
    scanf("%lf", pf);
}
```

프로그래밍 연습 소스 03

```c
void show_info(int mode, double distance, double fuel)
{
    printf("Fuel consumption is ");
    if (mode == METRIC)
        printf("%.2f liters per 100 km.\n", 100 * fuel / distance);
    else
        printf("%.1f miles per gallon.\n", distance / fuel);
}
```

프로그래밍 연습 소스 03

```c
// compile with pe12-3a.c
#include <stdio.h>
#include "pe12-3a.h"
int main(void)
{
    int mode;
    int prev_mode = METRIC;
    double distance, fuel;
    printf("Enter 0 for metric mode, 1 for US mode: ");
    scanf("%d", &mode);
```

프로그래밍 연습 소스 03

```
11      while (mode >= 0)
12      {
13          check_mode(&mode);
14          if (mode == USE_RECENT)
15              mode = prev_mode;
16          prev_mode = mode;
17          get_info(mode, &distance, &fuel);
18          show_info(mode, distance, fuel);
19          printf("Enter 0 for metric mode, 1 for US mode");
20          printf(" (-1 to quit): ");
21          scanf("%d", &mode);
22      }
23      printf("Done.\n");
24      return 0;
25  }
```

프로그래밍 연습 소스 03

```
Enter 0 for metric mode, 1 for US mode: 0
Enter distance traveled in kilometers: 600
Enter fuel consumed in liters: 78.8
Fuel consumption is 13.13 liters per 100 km.
Enter 0 for metric mode, 1 for US mode (-1 to quit): 13.13
Invalid mode specified. Mode 0(Metric) used.
Enter distance traveled in kilometers: Enter fuel consumed in liters: 1
Fuel consumption is 769.23 liters per 100 km.
Enter 0 for metric mode, 1 for US mode (-1 to quit): 434
Invalid mode specified. Mode 0(Metric) used.
Enter distance traveled in kilometers: 12.7
Enter fuel consumed in liters: 54.2
Fuel consumption is 426.77 liters per 100 km.
Enter 0 for metric mode, 1 for US mode (-1 to quit): 3
Invalid mode specified. Mode 0(Metric) used.
Enter distance traveled in kilometers: 388
Enter fuel consumed in liters: 15.3
Fuel consumption is 3.94 liters per 100 km.
Enter 0 for metric mode, 1 for US mode (-1 to quit): -1
Done.
```

프로그래밍 연습 문제

p663

번에 있는 설명을 참조하라.

04 자신이 호출된 횟수를 리턴하는 함수를 작성하고,
루프로 테스트하라.

프로그래밍 연습 소스 04

```c
#include <stdio.h>
int callme(void);
int main(void)
{
    int i, lim, count;
    printf("Enter number of loops: ");
    scanf("%d", &lim);
    for (i = 0; i < lim; i++)
        callme();
    count = callme();
    printf("Function was called %d times in loop and once afterwards.
        count - 1);
    return 0;
}
```

프로그래밍 연습 소스 04

```
16 /* keeps track of the number of times it was called */
17 int callme(void)
18 {
19     static int called = 0;
20     return ++called;
21 }
```

```
Enter number of loops: 3
Function was called 3 times in loop and once afterwards.
```

witch를 사용하여 연습

04

프로그래밍 연습 문제

p664

를 다시 작성하라.

05

1에서 10까지 범위의 난수를 100개 발생시켜 내림차순으로 정렬하여 출력하는 프로그램을 작성하라.

('11장: 문자열과 문자열 함수'에서 보았던 정렬 알고리즘을 int형에 적용할 수 있다.

이 경우에는 수들을 정렬하면 된다.)

프로그래밍 연습 소스 05

```c
#include <stdio.h>
#include <stdlib.h>
void print(const int array[], int limit);
void sort(int array[], int limit);
#define SIZE 100
int main(void)
{
    int i;
    int arr[SIZE];
    for (i = 0; i < SIZE; i++)
        arr[i] = rand() % 10 + 1; // 1 ~ 10 난수 발생
    puts("initial array");
    print(arr, SIZE);
    sort(arr, SIZE);
    puts("\nsorted array");
    print(arr, SIZE);
    return 0;
}
```

프로그래밍 연습 소스 05

```c
/* sort.c -- sorts an integer array in decreasing order */
void sort(int array[], int limit)
{
    int top, search, temp;

    for (top = 0; top < limit - 1; top++)
        for (search = top + 1; search < limit; search++)
            if (array[search] > array[top])
            {
                temp = array[search];
                array[search] = array[top];
                array[top] = temp;
            }
}
```

프로그래밍 연습 소스 05

```c
/* print.c -- prints an array */
void print(const int array[], int limit)
{
    int index;
    for (index = 0; index < limit; index++)
    {
        printf("%2d ", array[index]);
        if (index % 10 == 9)
            putchar('\n');
    }
    if (index % 10 != 0) // if last line not complete
        putchar('\n');
}
```

프로그래밍 연습 소스 05

```
initial array
 2  8  5  1 10  5  9  9  3  5
 6  6  2  8  2  2  6  3  8  7
 2  5  3  4  3  3  2  7  9  6
 8  7  2  9 10  3  8 10  6  5
 4  2  3  4  4  5  2  2  4  9
 8  5  3  8  8 10  4  2 10  9
 7  6  1  3  9  7  1  3  5  9
 7  6  1 10  1  1  7  2  4  9
10  4  5  5  7  1  7  7  2  9
 5 10  7  4  8  9  9  3 10  2

sorted array
10 10 10 10 10 10 10 10 10  9
 9  9  9  9  9  9  9  9  9  9
 9  8  8  8  8  8  8  8  8  8
 7  7  7  7  7  7  7  7  7  7
 7  6  6  6  6  6  6  6  5  5
 5  5  5  5  5  5  5  5  5  4
 4  4  4  4  4  4  4  4  3  3
 3  3  3  3  3  3  3  3  3  2
 2  2  2  2  2  2  2  2  2  2
 2  2  2  1  1  1  1  1  1  1
```

프로그래밍 연습 문제 p664

06

1에서 10까지 범위의 난수를 1,000개 발생시키는 프로그램을 작성하라.
난수들을 저장하거나 출력하지 마라.
다만 각 난수가 몇 번씩 발생하는지 출력하라.
서로 다른 10개의 종자값으로 프로그램을 테스트하라.
각 난수들이 같은 횟수로 발생하는가?
이 장에서 배운 함수들을 사용하거나,
같은 형식을 따르는 ANSI C의 rand()와 srand() 함수를 사용할 수 있다.
이것은 난수 발생기의 무작위성을 테스트하는 한 가지 방법이다.

프로그래밍 연습 소스 06

```c
#include <stdio.h>
#include <stdlib.h>
void print(const int array[], int limit);
#define LIMIT 1000
#define DIGITS 10
#define SEEDS 10
int main(void)
{
    int i;
    int counts[DIGITS];
    int cumul[DIGITS] = { 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    int seed;
    int val;
    printf("Seed ");
    for (i = 1; i <= DIGITS; i++)
        printf("%3d ", i);
    putchar('\n');
```

프로그래밍 연습 소스 06

```
19      for (seed = 1; seed <= SEEDS; seed++)
20      {
21          srand(seed);
22          for (i = 0; i < DIGITS; i++)
23              counts[i] = 0;
24          for (i = 0; i < LIMIT; i++)
25          {
26              val = rand() % 10 + 1;
27              counts[val - 1]++;
28              cumul[val - 1]++;
29          }
30          printf("%3d ", seed);
31          print(counts, DIGITS);
32      }
```

프로그래밍 연습 소스 06

```
    printf("Average for %d seeds\n", SEEDS);
    for (i = 0; i < DIGITS; i++)
        cumul[i] /= SEEDS;
    printf("ave ");
    print(cumul, DIGITS);
    return 0;
}
```

프로그래밍 연습 소스 06

```c
/* print.c -- prints an array */
void print(const int array[], int limit)
{
    int index;
    for (index = 0; index < limit; index++)
    {
        printf("%3d ", array[index]);
        if (index % 10 == 9)
            putchar('\n');
    }
    if (index % 10 != 0) // if last line not complete
        putchar('\n');
}
```

Seed	1	2	3	4	5	6	7	8	9	10
1	96	101	103	116	96	109	89	87	109	94
2	102	98	95	114	101	100	95	93	96	106
3	94	108	102	106	98	92	105	95	101	99
4	94	103	106	86	89	105	101	110	101	105
5	108	90	105	95	100	123	90	85	94	110
6	124	92	96	94	113	105	101	99	75	101
7	103	91	91	110	111	105	107	95	100	87
8	103	104	87	96	101	121	89	97	100	102
9	98	91	103	101	110	84	109	101	96	107
10	105	92	107	108	99	90	98	93	100	108

Average for 10 seeds

ave	102	97	99	102	101	103	98	95	97	101

07

리스트 12.13의 출력을 보여 준 뒤에 논의한 내용처럼 행동하는 프로그램을 작성하라. 즉, 프로그램의 출력이 다음과 같아야 한다.

```
던질 횟수를 입력하시오(끝내려면 q).
18
주사위 면 수와 던질 주사위 개수를 입력하시오(끝내려면 0).
6 3
6면 주사위 3개를 18번 던져 얻은 결과는 다음과 같습니다.
12 10 6 9 8 14 8 15 9 14 12 17 11 7 10
13 8 14
던질 횟수를 입력하시오(끝내려면 q).
q
행운을 빕니다!
```

프로그래밍 연습 소스 07

```c
#include <stdio.h>
#include <stdlib.h> /* for srand() */
#include <time.h> /* for time() */
int rollem(int);
int main(void)
{
    int dice, count, roll;
    int sides;
    int set, sets;
    srand((unsigned int)time(0)); /* randomize rand() */
    printf("Enter the number of sets; enter q to stop: ");
    while (scanf("%d", &sets) == 1)
    {
        printf("How many sides and how many dice? ");
        if (scanf("%d %d", &sides, &dice) != 2)
        {
            puts("not integers -- terminating input loop.");
            break;
        }
```

```
20          printf("Here are %d sets of %d %d-sided throws.\n", sets, dice, sides);
21          for (set = 0; set < sets; set++)
22          {
23              for (roll = 0, count = 0; count < dice; count++)
24                  roll += rollem(sides);
25              /* running total of dice pips */
26              printf("%4d ", roll);
27              if (set % 15 == 14)
28                  putchar('\n');
29          }
30          if (set % 15 != 0)
31              putchar('\n');
32          printf("How many sets? Enter q to stop: ");
33      }
34      puts("GOOD FORTUNE TO YOU!\n");
35      return 0;
36  }
```

```c
int rollem(int sides)
{
    int roll;
    roll = rand() % sides + 1;
    return roll;
}
```

```
Enter the number of sets; enter q to stop: 18
How many sides and how many dice? 6 3
Here are 18 sets of 3 6-sided throws.
  16   11   13   11    7   10   11    6   13    5    9    7    6    8   13
  11   10   11
How many sets? Enter q to stop: Q
GOOD FORTUNE TO YOU!
```

08 다음은 어떤 프로그램의 일부이다.

```
// pe12-8.c
#include <stdio.h>
int * make_array(int elem, int val);
void show_array(const int ar[], int n);
int main(void)
{
    ...................
    return 0;
}
```

프로그래밍 연습 문제 p665

08

make_array()와 show_array()의 함수 정의를 작성하여
이 프로그램을 완성하라.
make_array() 함수는 두 개의 전달인자를 사용한다.
첫 번째 전달인자는 int형 배열의 원소 개수를 나타내고,
두 번째 전달인자는 각 원소에 대입할 하나의 값을 나타낸다.
이 함수는 malloc()을 사용하여 적절한 크기의 배열을 생성하고,
지정된 값으로 각 원소들을 설정하고, 그 배열을 가리키는 포인터를 리턴한다.
show_array() 함수는 배열의 내용을 출력하는데,
수들을 한 라인에 8개씩 출력한다.

프로그래밍 연습 소스 08

```c
#include <stdio.h>
#include <stdlib.h>
int * make_array(int elem, int val);
void show_array(const int ar[], int n);
int main(void)
{
    int * pa;
    int size;
    int value;
    printf("Enter the number of elements: ");
```

프로그래밍 연습 소스 08

```
11      while (scanf("%d", &size) == 1 && size > 0)
12      {
13          printf("Enter the initialization value: ");
14          scanf("%d", &value);
15          pa = make_array(size, value);
16          if (pa)
17          {
18              show_array(pa, size);
19              free(pa);
20          }
21          printf("Enter the number of elements (<1 to quit): ");
22      }
23      printf("Done.\n");
24      return 0;
25  }
```

프로그래밍 연습 소스 08

```c
int * make_array(int elem, int val)
{
    int i;
    int * pt = (int *)malloc(elem * sizeof(int));
    if (pt)
        for (i = 0; i < elem; i++)
            pt[i] = val;
    return pt;
}
```

프로그래밍 연습 소스 08

```c
37 void show_array(const int ar[], int n)
38 {
39     int i;
40     for (i = 0; i < n; i++)
41     {
42         printf("%6d ", ar[i]);
43         if (i % 8 == 7)
44             putchar('\n');
45     }
46     if (i % 8 != 0)
47         putchar('\n');
48 }
```

```
Enter the number of elements: 5
Enter the initialization value: 5
     5      5      5      5      5
Enter the number of elements (<1 to quit): 2
Enter the initialization value: 9
     9      9
Enter the number of elements (<1 to quit): -1
Done.
```

프로그래밍 연습 문제 p665

09

다음과 같이 행동하는 프로그램을 작성하라.
먼저, 입력하려는 단어가 몇 개인지 물어본다.
그리고 나서 사용자가 단어를 입력하게 하고,
그런 다음 단어를 화면상에 나타낸다.
malloc()과 첫 번째 질문(단어 수)에 맞는 답을 사용해
상응하는 수의 char를 가리키는 포인터(pointer-to-char)의
동적 배열을 만든다.

프로그래밍 연습 문제 p665

09

다음과 같이 행동하는 프로그램을 작성하라.
먼저, 입력하려는 단어가 몇 개인지 물어본다.
그리고 나서 사용자가 단어를 입력하게 하고,
그런 다음 단어를 화면상에 나타낸다.
malloc()과 첫 번째 질문(단어 수)에 맞는 답을 사용해
상응하는 수의 char를 가리키는 포인터(pointer-to-char)의
동적 배열을 만든다.

09

(배열의 각 원소가 char를 가리키는 포인터(pointer-to-char)이기 때문에, malloc()의 리턴 값을 저장하는데 사용된 포인터는 char를 가리키는 포인터(pointers-to-a-pointer-to-char)를 가리키는 포인터이어야 함에 유념하라)

문자열을 읽을 때, 프로그램은 단어에 char의 임시 배열의 의미를 부여하여야 하며, malloc()를 사용해 단어를 유지하는데 충분한 저장 공간을 할당하고, char 포인터의 배열에 주소를 저장해야 한다.

이렇게, 문자 포인터의 배열 만들기를 마치고 각 포인터들은 특정 단어를 지정하기에 딱 알맞은 크기에 객체를 가리킨다.

프로그래밍 연습 문제

p666

09

```
How many words do you wish to enter? 5
Enter 5 words now:
I enjoyed doing this exerise
Here are your words:
I
enjoyed
doing
this
exercise
```

프로그래밍 연습 소스 09

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 31
int main(void)
{
    int words, i;
    char temp[MAX];
    char ** ptr_str;
    printf("How many words do you wish to enter? ");
    scanf("%d", &words);

    ptr_str = (char **)malloc(words * sizeof(char *));
```

프로그래밍 연습 소스 09

```c
    printf("Enter %d words now:\n", words);
    for (i = 0; i < words; i++)
    {
        scanf("%30s", temp);
        ptr_str[i] = (char *)malloc(strlen(temp) + 1);
        strcpy(ptr_str[i], temp);
    }

    puts("Here are your words:");
    for (i = 0; i < words; i++)
    {
        puts(ptr_str[i]);
    }
    return 0;
}
```

```
How many words do you wish to enter? 3
Enter 3 words now:
vision
happy
Dream
Here are your words:
vision
happy
Dream
```

KeyPoint 핵심 개념

- int *pt ;
- pt = (int *)malloc(max * sizeof(int);
- free(pt)

이번 학습에 이어!

- 다음 강의 내용
  - 제 13장 파일 입출력, 연습문제

Thank you !