

Basic Computer Programming

Lecture 12

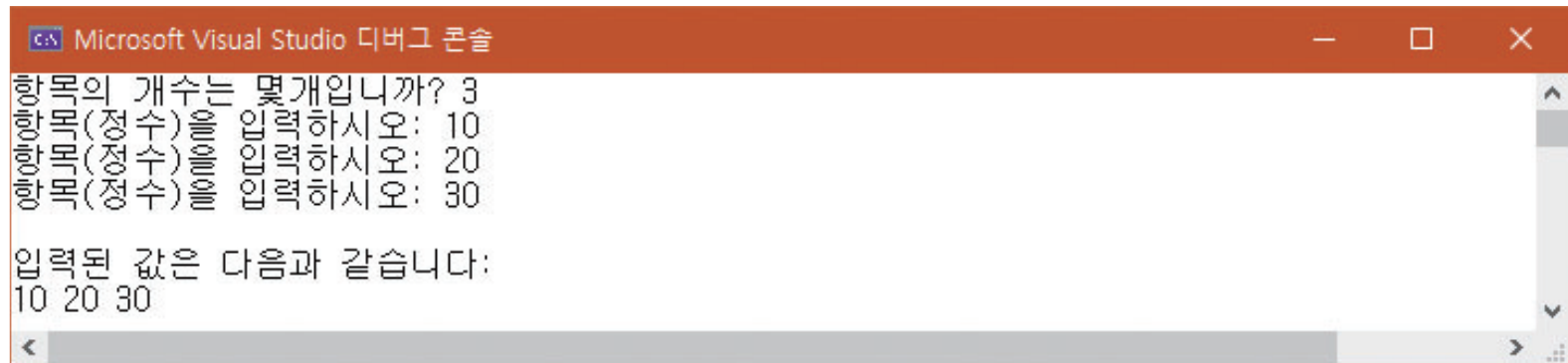
Electrical & Electronics Engineering
Chung-Ang University

Contents

- Understand the concept of dynamic memory allocation.
- Functions related to dynamic memory allocation can be used.

The programs we will make in this chapter

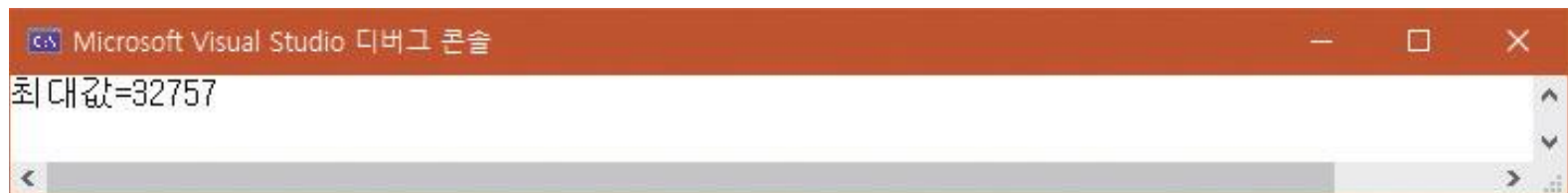
- Ask the user about the size of the desired array, and dynamically create an array of that size.



```
Microsoft Visual Studio 디버그 콘솔
항목의 개수는 몇개입니까? 3
항목(정수)을 입력하시오: 10
항목(정수)을 입력하시오: 20
항목(정수)을 입력하시오: 30

입력된 값은 다음과 같습니다:
10 20 30
```

- Create a dynamic array with a size of 1000, fill it with random numbers, and then calculate the maximum value.



```
Microsoft Visual Studio 디버그 콘솔
최대값=32757
```

Concepts of Dynamic Memory Allocation

- How programs are allocated memory
- Static
- Dynamic



Static Memory allocation

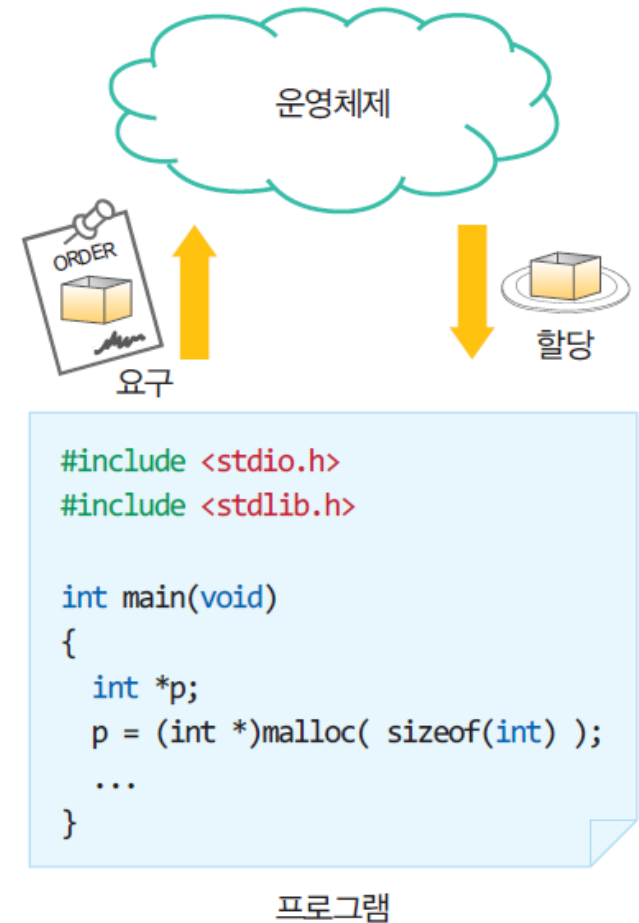
- Allocate a predetermined size of memory before the program starts
- Determine the size of the memory before the program starts

```
int sarray[10];
```

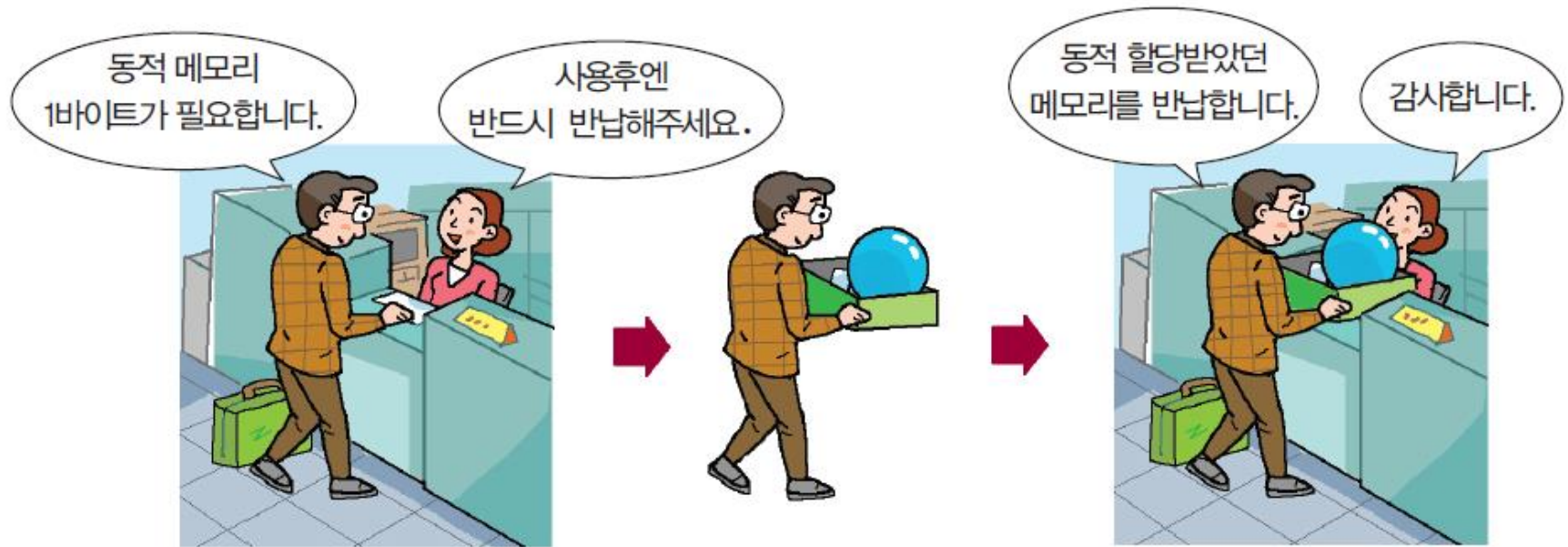
- Fail to process if an input larger than the initial determined size comes in
- Waste remaining memory space if smaller inputs come in

Dynamic Memory Allocation

- Dynamic allocation of memory during execution
- Return memory to the system after use
- Allocate as much as you need and use memory very efficiently
- Use library functions from the malloc() family



Dynamic Memory Allocation Process



malloc()

Syntax 13.1 malloc()의 형식

형식 `void *malloc(size_t size);` // `size_t`는 부호 없는 정수타입이다.

예 `#include <stdlib.h>`

`char *p;`
`p = (char *)malloc(100);`
`...`

설명 `size` 바이트만큼의 동적 메모리를 할당한다.

free()

Syntax 13.2 free()의 형식

형식 `void free(void *ptr)`

예 `#include <stdlib.h>`

```
char *p;  
p = (char *)malloc(100);  
...  
free(p);
```

설명 ptr이 가리키는 동적 메모리 블록을 반납한다.

Example #1

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

int main(void)
{
    char* pc;

    pc = (char*)malloc(1 * sizeof(char));

    *pc = 'a';
    printf("%c \n", *pc);
    free(pc);

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

a

Example #2

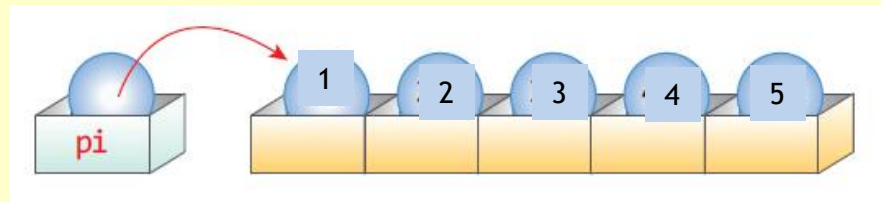
```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int* pi;

    pi = (int*)malloc(5 * sizeof(int));

    *pi = 1; // pi[0] = 1;
    *(pi + 1) = 2; // pi[1] = 2;
    *(pi + 2) = 3; // pi[2] = 3;
    *(pi + 3) = 4; // pi[3] = 4;
    *(pi + 4) = 5; // pi[4] = 5;

    free(pi);
    return 0;
}
```



Interim check

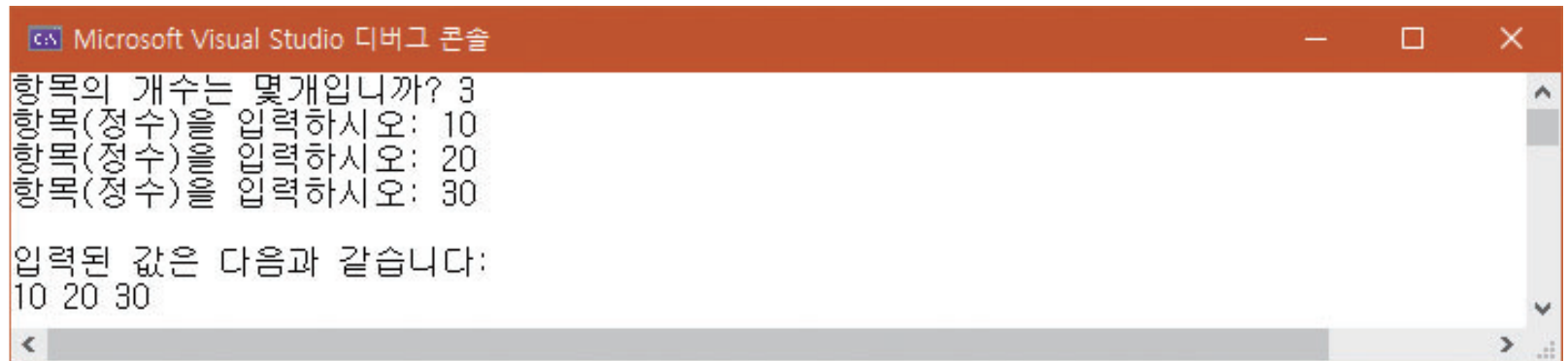


중간점검

1. 프로그램의 실행 도중에 메모리를 할당받아서 사용하는 것을 _____ 이라고 한다.
2. 동적으로 메모리를 할당받을 때 사용하는 대표적인 함수는 _____ 이다.
3. 동적으로 할당된 메모리를 해제하는 함수는 _____ 이다.
4. 동적 메모리 함수의 원형은 헤더파일 _____ 에 정의되어 있다.

Lab: create an array of sizes that the user enters.

- In this practice, let's ask the user how many items they want and dynamically create an array of that size.



```
Microsoft Visual Studio 디버그 콘솔
항목의 개수는 몇개입니까? 3
항목(정수)을 입력하시오: 10
항목(정수)을 입력하시오: 20
항목(정수)을 입력하시오: 30

입력된 값은 다음과 같습니다:
10 20 30
```

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

```
int main(void)
```

```
{
```

```
    int* p;
```

```
    int i, items;
```

```
    printf("항목의 개수는 몇개입니까? ");
```

```
    scanf("%d", &items);
```

```
    p = (int*)malloc(sizeof(int) * items);
```

```
    for (i = 0; i < items; i++) {
```

```
        printf("항목(정수)을 입력하십시오: ");
```

```
        scanf("%d", &p[i]);
```

```
    }
```

```
    printf("\n입력된 값은 다음과 같습니다: \n");
```

```
    for (i = 0; i < items; i++)
```

```
        printf("%d ", p[i]);
```

```
    printf("\n");
```

```
    free(p);
```

```
    return 0;
```

```
}
```

Lab: fill the dynamic array with random numbers.

- Let's create a dynamic array with a size of 1000 and fill the dynamic array with random numbers. The maximum value of the elements in the dynamic array is calculated and output.



Sol:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 1000

int main(void)
{

    int* p = NULL;
    int i = 0;

    p = (int*)malloc(SIZE * sizeof(int));
    if (p == NULL) {
        printf("메모리 할당 오류\n");

        exit(1);
    }

    for (i = 0; i < SIZE; i++)
        p[i] = rand();
```


Sol:

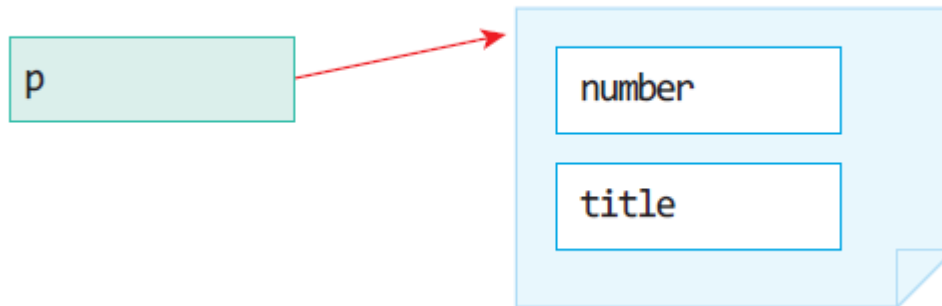
```
int max = p[0];
    for (i = 1; i < SIZE; i++) {
        if (p[i] > max)
            max = p[i];
    }

    printf("최대값=%d \n", max);
    free(p);

    return 0;
}
```

Let's dynamically create a structure

```
struct Book {  
    int number;  
    char title[100];  
};  
  
struct Book* p;  
p = (struct Book*)malloc(2 * sizeof(struct Book));
```



Example

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

struct Book {
    int number;
    char title[100];
};

int main(void)
{
    struct Book* p;

    p = (struct Book*)malloc(2 * sizeof(struct Book));

    if (p == NULL) {
        printf("메모리 할당 오류\n");
        exit(1);
    }
}
```

Example

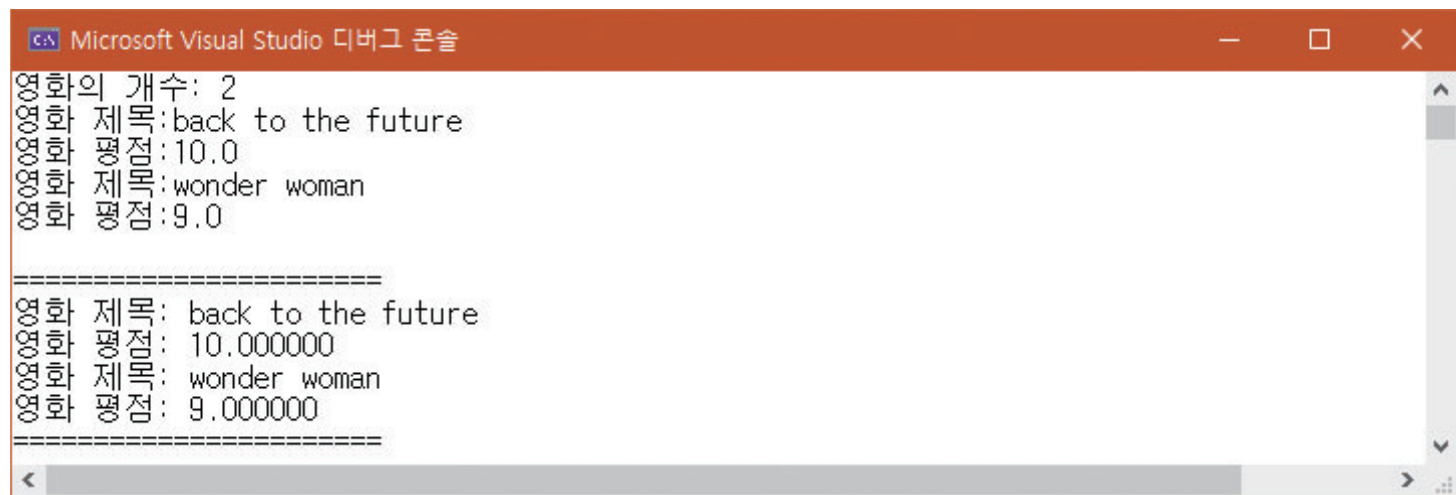
```
p[0].number = 1; // (*p).number = 1
strcpy(p[0].title, "C Programming");

p[1].number = 2; // (*p+1).number = 2
strcpy(p[1].title, "Data Structure");

free(p);
return 0;
}
```

Lab: Dynamic Structural Array

- Organize all movies by making an array of struct. Let's take the number of movies from the user and dynamically create an array of structures and store all the information here.



The screenshot shows the Microsoft Visual Studio 디버그 콘솔 (Debug Console) window. It displays the following text:

```
영화의 개수: 2
영화 제목: back to the future
영화 평점: 10.0
영화 제목: wonder woman
영화 평점: 9.0

=====
영화 제목: back to the future
영화 평점: 10.000000
영화 제목: wonder woman
영화 평점: 9.000000
=====
```

The output shows the movie titles and ratings entered by the user, followed by a separator line and the same data displayed with floating-point formatting (10.000000 and 9.000000).

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

// 영화를 구조체로 표현
struct movie
{
    char title[100]; // 영화 제목
    double rating; // 영화 평점
};

int main(void)
{
    struct movie* ptr;
    int i, n;

    printf("영화의 개수: ");
    scanf("%d", &n);
    getchar(); // 줄바꿈 문자 제거

    ptr = (struct movie*)malloc(n * sizeof(struct movie));
    if (ptr == NULL) {
        printf("메모리 할당 오류\n");
        exit(1);
    }
}
```

Sol:

```
for (i = 0; i < n; i++) {  
    printf("영화 제목:");  
    gets_s(ptr[i].title, 100); // 중간에 공백이 있는 제목도 받을 수도 있어야  
    한다.  
    printf("영화 평점:");  
    scanf("%lf", &ptr[i].rating);  
    getchar(); // 줄바꿈 문자 제거  
}
```

```
printf("\n===== \n");
```

```
for (i = 0; i < n; i++) {  
    printf("영화 제목: %s \n", ptr[i].title);  
    printf("영화 평점: %lf \n", ptr[i].rating);  
}
```

```
printf("===== \n");
```

```
free(ptr);
```

```
return 0;
```

```
}
```