



C프로그래밍

Lecture 11. 1차원 배열

동덕여자대학교
데이터사이언스 전공
권 범

목차

- ❖ 01. 배열의 이해와 배열의 선언 및 초기화 방법
- ❖ 02. 배열을 이용한 문자열 변수의 표현
- ❖ 03. 연습 문제

01. 배열의 이해와 배열의 선언 및 초기화 방법

02. 배열을 이용한 문자열 변수의 표현

03. 연습 문제

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 배열이란 무엇인가? (1/2)

상황 가정

- ✓ 동덕 빌딩에 사는 사람들의 가구별 가족 수를 기록하는 프로그램을 만든다고 가정해 보겠습니다.
- ✓ 동덕 빌딩은 10층까지 있고, 지하 1층을 포함해서 총 11층으로 이뤄져 있습니다.
- ✓ 각 층에는 한 가구씩 살고 있습니다.

가구별 가족 수를 저장하기 위해서는
총 11개의 변수를 선언해야 할 것입니다.

```
int main(void)
{
    int floor0, floor1, floor2, floor3, floor4, floor5;    // floor0은 지하 1층
    int floor6, floor7, floor8, floor9, floor10;
    ...
}
```

다수의 정보를 저장하기 위해서는
다수의 변수를 선언해야 합니다.

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 배열이란 무엇인가? (2/2)

- 만약 동덕 빌딩이 20층까지 있고, 층마다 10가구씩 살고 있다고 하면 어떻게 될까요?
- 가구별 가족 수를 기록하는 프로그램을 제작하기 위해서 변수 200개를 선언해야 할 것입니다.

경우에 따라서 다수의 변수를 선언하는 것은
매우 번거로울 수 있습니다.

- ✓ 다수의 변수 선언을 용이하게 하기 위해서 배열이라는 것이 제공됩니다.
- ✓ 배열을 이용하면 하나의 선언을 통해서 둘 이상의 변수를 선언할 수 있습니다.

- ✓ 배열은 단순히 다수의 변수 선언을 대신하지 않습니다.
- ✓ 다수의 변수로는 할 수 없는 일을 배열을 선언하면 할 수 있습니다.

- ✓ 배열은 1차원의 형태로도 2차원의 형태로도 선언할 수 있습니다.
- ✓ 이번 수업에서는 1차원 형태의 배열에 대해서 학습합니다.

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 1차원 배열 선언에 필요한 것 세 가지 (1/2)

● 1차원 배열 선언의 예

```
int oneDimArr[4];
```

int	배열을 이루는 요소(변수)의 자료형
oneDimArr	배열의 이름
[4]	배열의 길이

생성되는 배열의 형태



```
int arr1[7];           // 길이가 7인 int형 1차원 배열 arr1
float arr2[10];         // 길이가 10인 float형 1차원 배열 arr2
double arr3[12];        // 길이가 12인 double형 1차원 배열 arr3
```

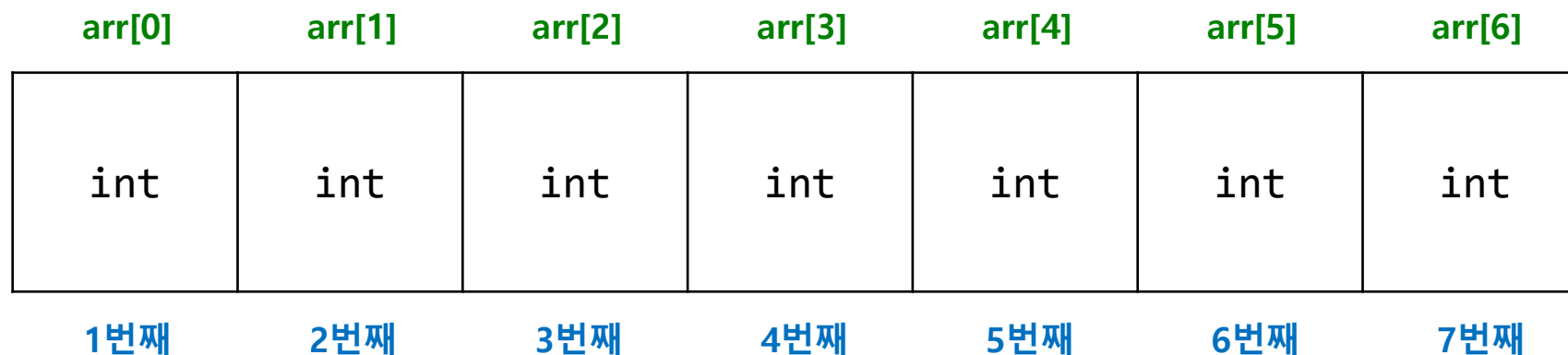
다양한 배열 선언의 예

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 1차원 배열 선언에 필요한 것 세 가지 (2/2)

- “int arr[7]” 선언에 의한 메모리 할당

```
int arr[7];
```



int형 정수 저장을 위한 4바이트 메모리 블록이 7개 존재하므로 총 28바이트가 할당 됩니다.

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 배열의 길이 선언은 ONLY! 상수

```
int main(void)
{
    int size = 10;
    int arr[size];    // 컴파일 오류 발생, size가 변수이므로
    return 0;
}
```



abc	E0028	식에 상수 값이 있어야 합니다.
✗	C2057	상수 식이 필요합니다.
✗	C2466	상수 크기 0의 배열을 할당할 수 없습니다.
✗	C2133	'arr': 알 수 없는 크기입니다.

배열의 길이를 선언할 때는 반드시 상수를 사용해야 합니다.

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 선언된 1차원 배열의 접근 (1/2)

```
arr[0] = 10;           // 배열 arr의 첫 번째 요소에 10을 저장해라!  
arr[1] = 12;           // 배열 arr의 두 번째 요소에 12을 저장해라!  
arr[2] = 25;           // 배열 arr의 세 번째 요소에 25을 저장해라!
```

1차원 배열 접근의 예

일반화

```
arr[idx] = 20;          // 배열 arr의 idx+1번째 요소에 20을 저장해라!
```

첫 번째 배열 요소의 인덱스(Index)는 0입니다.

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 선언된 1차원 배열의 접근 (2/2)

```
1  /* array1.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int arr[5];
7      int sum = 0, j;
8
9      arr[0] = 10, arr[1] = 20, arr[2] = 30, arr[3] = 40, arr[4] = 50;
10
11     for (j = 0; j < 5; j++)
12         sum = sum + arr[j];
13
14     printf("배열 요소에 저장된 값의 합: %d\n", sum);
15     return 0;
16 }
```

배열 요소에 저장된 값의 합: 150

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 배열! 선언과 동시에 초기화하기 (1/3)

```
int arr1[5] = {1, 2, 3, 4, 5};
```

초기화 리스트로 초기화



순서대로 초기화

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 배열! 선언과 동시에 초기화하기 (2/3)

```
int arr2[] = {1, 2, 3, 4, 5, 6, 7};
```

초기화 리스트는 존재하고
배열의 길이 정보가 생략된 경우



```
int arr2[7] = {1, 2, 3, 4, 5, 6, 7};
```

컴파일러가 배열의 길이 정보를 채웁니다.

초기화 리스트 항목의 수를 통해
배열 크기를 결정합니다.

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 배열! 선언과 동시에 초기화하기 (3/3)

```
int arr3[5] = {1, 2};
```

초기화 값이 부족한 경우



부족한 부분은 0으로 채워짐

01. 배열의 이해와 배열의 선언 및 초기화 방법

❖ 1차원 배열의 선언, 초기화 및 접근 관련 예제

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int arr1[5] = {1, 2, 3, 4, 5};
5      int arr2[] = {1, 2, 3, 4, 5, 6, 7};
6      int arr3[5] = {1, 2};
7      int ar1Len, ar2Len, ar3Len, j;
8
9      printf("배열 arr1의 크기: %d\n", sizeof(arr1));
10     printf("배열 arr2의 크기: %d\n", sizeof(arr2));
11     printf("배열 arr3의 크기: %d\n", sizeof(arr3));
12
13     ar1Len = sizeof(arr1) / sizeof(int);
14     ar2Len = sizeof(arr2) / sizeof(int);
15     ar3Len = sizeof(arr3) / sizeof(int);
16
17     for (j = 0; j < ar1Len; j++)
18         printf("%d ", arr1[j]);
19     printf("\n");
```

```
20         for (j = 0; j < ar2Len; j++)
21             printf("%d ", arr2[j]);
22         printf("\n");
23
24         for (j = 0; j < ar3Len; j++)
25             printf("%d ", arr3[j]);
26         printf("\n");
27
28         return 0;
29     }
```

배열 arr1의 크기: 20
배열 arr2의 크기: 28
배열 arr3의 크기: 20

```
1 2 3 4 5
1 2 3 4 5 6 7
1 2 0 0 0
```

02. 배열을 이용한 문자열 변수의 표현

- 01. 배열의 이해와 배열의 선언 및 초기화 방법
- 03. 연습 문제

02. 배열을 이용한 문자열 변수의 표현

❖ char형 배열의 문자열 저장과 널(Null) 문자 (1/3)

```
char str[14] = "Good morning!";
```

배열에 문자열 저장




문자열의 끝에 널 문자라고 불리는 `\0`가 삽입되었음에 주목하세요.
널 문자는 문자열의 끝을 의미합니다.

문자열을 변수로 선언하는데 있어서
사용하게 되는 것이 바로 배열입니다.

02. 배열을 이용한 문자열 변수의 표현

❖ char형 배열의 문자열 저장과 널(Null) 문자 (2/3)

```
1  #include <stdio.h>
2  int main(void)
3  {
4      char str1[5] = "Good";
5      char str2[] = "morning";
6
7      printf("%s\n", str1);
8      printf("%s %s\n", str1, str2);
9
10     return 0;
11 }
```



Good
Good morning

str1[0] str1[1] str1[2] str1[3] str1[4]

G	o	o	d	\0
---	---	---	---	----

str2[0] str2[1] str2[2] str2[3] str2[4] str2[5] str2[6] str2[7]

m	o	r	n	i	n	g	\0
---	---	---	---	---	---	---	----

02. 배열을 이용한 문자열 변수의 표현

❖ char형 배열의 문자열 저장과 널(Null) 문자 (3/3)

```
1  #include <stdio.h>
2  int main(void)
3  {
4      char str[] = "Good morning!";
5
6      printf("배열 str의 크기: %d\n", sizeof(str));
7      printf("널 문자 문자형 출력: %c\n", str[13]);
8      printf("널 문자 정수형 출력: %d\n", str[13]);
9
10     str[12]='?';    // 배열 str에 저장된 문자열 데이터는 변경 가능!
11
12     printf("문자열 출력: %s\n", str);
13
14     return 0;
15 }
```

배열 str의 크기: 14
널 문자 문자형 출력:
널 문자 정수형 출력: 0
문자열 출력: Good morning?


널 문자('\0')는 아스키 코드 값이 0입니다.

str[0]	str[1]	str[2]	str[3]	str[4]	str[5]	str[6]	str[7]	str[8]	str[9]	str[10]	str[11]	str[12]	str[13]
G	o	o	d		m	o	r	n	i	n	g	!	\0

02. 배열을 이용한 문자열 변수의 표현

❖ 널 문자와 공백 문자의 비교

```
1  #include <stdio.h>
2  int main(void)
3  {
4      char nu = '\0';          // 널 문자 저장
5      char sp = ' ';          // 공백 문자 저장
6
7      printf("%d %d\n", nu, sp); // 0과 32 출력
8      return 0;
9  }
```



0 32

- ✓ 널 문자를 %c를 이용해서 출력 시 아무것도 출력되지 않습니다.
- ✓ 그렇다고 해서 널 문자가 공백 문자인 것은 아닙니다.
- ✓ 널 문자의 아스키 코드 값은 0이고, 공백 문자의 아스키 코드 값은 32입니다.
- ✓ 널 문자는 모니터 출력에서 의미를 갖지 않습니다.
- ✓ 그래서 아무것도 출력이 되지 않을 뿐입니다.

02. 배열을 이용한 문자열 변수의 표현

❖ scanf 함수를 이용한 문자열의 입력 (1/2)

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  int main(void)
4  {
5      char str[50];
6      int idx=0;
7
8      printf("문자열 입력: ");
9      scanf("%s", str);          // 문자열을 입력 받아서 배열 str에 저장!
10     printf("입력 받은 문자열: %s\n", str);
11
12     printf("문자 단위 출력: ");
13     while(str[idx] != '\0')
14     {
15         printf("%c", str[idx]);
16         idx++;
17     }
18     printf("\n");
19     return 0;
20 }
```

✓ scanf 함수를 이용해서 문자열 입력 시 서식 문자 %s를 사용합니다.

scanf("%s", str);

✓ 위와 같이 배열 이름 str의 앞에는 & 연산자를 붙이지 않습니다.

문자열 입력: Simple
입력 받은 문자열: Simple
문자 단위 출력: Simple

scanf 함수의 호출을 통해서 입력 받은 문자열의 끝에도
널 문자가 존재함을 확인하기 위한 문장

02. 배열을 이용한 문자열 변수의 표현

❖ scanf 함수를 이용한 문자열의 입력 (2/2)

```
char arr1[ ] = {'H', 'e', 'l', 'l', 'o'};  
char arr2[ ] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

- ✓ arr1은 문자열이 아닌 문자 배열입니다.
- ✓ 반면 arr2는 문자열입니다.

널 문자 \0 의 존재여부는 문자열의 판단 여부가 됩니다.

02. 배열을 이용한 문자열 변수의 표현

❖ 문자열의 끝에 널 문자가 필요한 이유 (1/2)

문자열의 시작은 판단할 수 있어도 문자열의 끝은 판단이 불가능합니다!
때문에 문자열의 끝을 판단할 수 있도록 널 문자가 삽입이 됩니다.

```
1  #include <stdio.h>
2  int main(void)
3  {
4      char str[50] = "I like C programming";
5      printf("string: %s\n", str);
6
7      str[8] = '\0';          // 9번째 요소에 널 문자 저장
8      printf("string: %s\n", str);
9
10     str[6] = '\0';          // 7번째 요소에 널 문자 저장
11     printf("string: %s\n", str);
12
13     str[1] = '\0';          // 2번째 요소에 널 문자 저장
14     printf("string: %s\n", str);
15     return 0;
16 }
```

- ✓ 배열의 시작 위치에 문자열이 저장되기 시작합니다.
- ✓ 따라서 시작 위치는 확인이 가능합니다.
- ✓ 하지만 배열의 끝이 문자열의 끝은 아니므로
널 문자가 삽입되지 않으면 문자열의 끝은 확인이
불가능합니다.

```
string: I like C programming
string: I like C
string: I like
string: I
```

02. 배열을 이용한 문자열 변수의 표현

❖ 문자열의 끝에 널 문자가 필요한 이유 (2/2)

문자열의 시작은 판단할 수 있어도 문자열의 끝은 판단이 불가능합니다!
때문에 문자열의 끝을 판단할 수 있도록 널 문자가 삽입이 됩니다.

```
1  #include <stdio.h>
2  int main(void)
3  {
4      char str[50] = "I like C programming";
5      printf("string: %s\n", str);
6
7      str[8] = '\0';          // 9번째 요소에 널 문자 저장
8      printf("string: %s\n", str);
9
10     str[6] = '\0';          // 7번째 요소에 널 문자 저장
11     printf("string: %s\n", str);
12
13     str[1] = '\0';          // 2번째 요소에 널 문자 저장
14     printf("string: %s\n", str);
15     return 0;
16 }
```

- ✓ 왼쪽 예제에서 보이듯이 printf 함수도 배열 str의 시작 위치를 기준으로 해서 널 문자를 만날 때까지 출력을 진행합니다.
- ✓ 따라서 널 문자가 없으면 printf 함수도 문자열의 끝을 알지 못합니다.

```
string: I like C programming
string: I like C
string: I like
string: I
```

02. 배열을 이용한 문자열 변수의 표현

❖ scanf 함수의 문자열 입력 특성

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  int main(void)
4  {
5      char str[50];
6      int idx=0;
7
8      printf("문자열 입력: ");
9      scanf("%s", str);          // 문자열을 입력 받아서 배열 str에 저장!
10     printf("입력 받은 문자열: %s\n", str);
11
12     printf("문자 단위 출력: ");
13     while(str[idx] != '\0')
14     {
15         printf("%c", str[idx]);
16         idx++;
17     }
18     printf("\n");
19     return 0;
20 }
```

앞서 보인 왼편의 예제를 실행할 때 다음과 같이 문자열을 입력하면 다음의 실행 결과를 보입니다.

문자열 입력: He is my friend
입력 받은 문자열: He
문자 단위 출력: He

- ✓ scanf 함수는 공백을 기준으로 데이터의 수를 구분합니다.
- ✓ 따라서 공백을 포함하는 문자열을 한 번의 scanf 함수 호출을 통해서 읽어 들이지는 못한다.

공백을 포함하는 문자열의 입력에 사용되는 함수는 이후에 별도로 설명합니다.

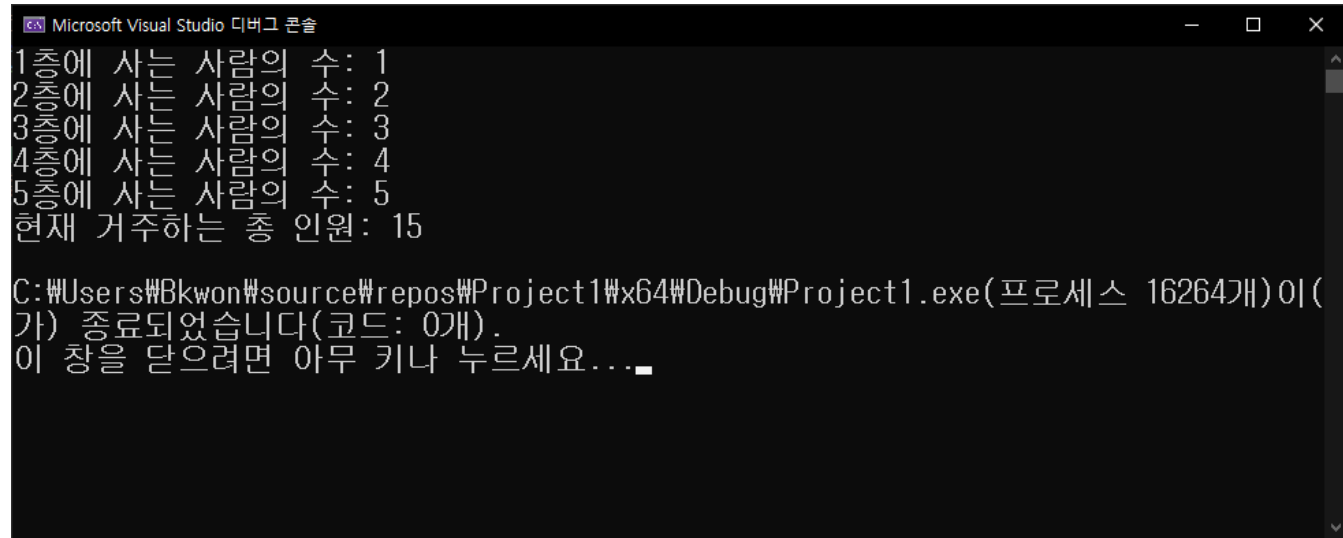
03. 연습 문제

- 01. 배열의 이해와 배열의 선언 및 초기화 방법
- 02. 배열을 이용한 문자열 변수의 표현

03. 연습 문제

❖ 연습 문제 1.

- 동덕 빌라는 5층으로 이뤄져 있습니다. 프로그램 실행 시 사용자로부터 1층부터 시작해서 5층까지 각 층에 사는 사람의 수를 입력 받습니다. 입력이 끝나면 동덕 빌라에 사는 전체 사람의 수를 계산해서 출력하는 프로그램을 작성해 보세요.



```
Microsoft Visual Studio 디버그 콘솔
1층에 사는 사람의 수: 1
2층에 사는 사람의 수: 2
3층에 사는 사람의 수: 3
4층에 사는 사람의 수: 4
5층에 사는 사람의 수: 5
현재 거주하는 총 인원: 15

C:\Users\Bkwon\source\repos\Project1\bin\x64\Debug\Project1.exe(프로세스 16264개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요....
```

03. 연습 문제

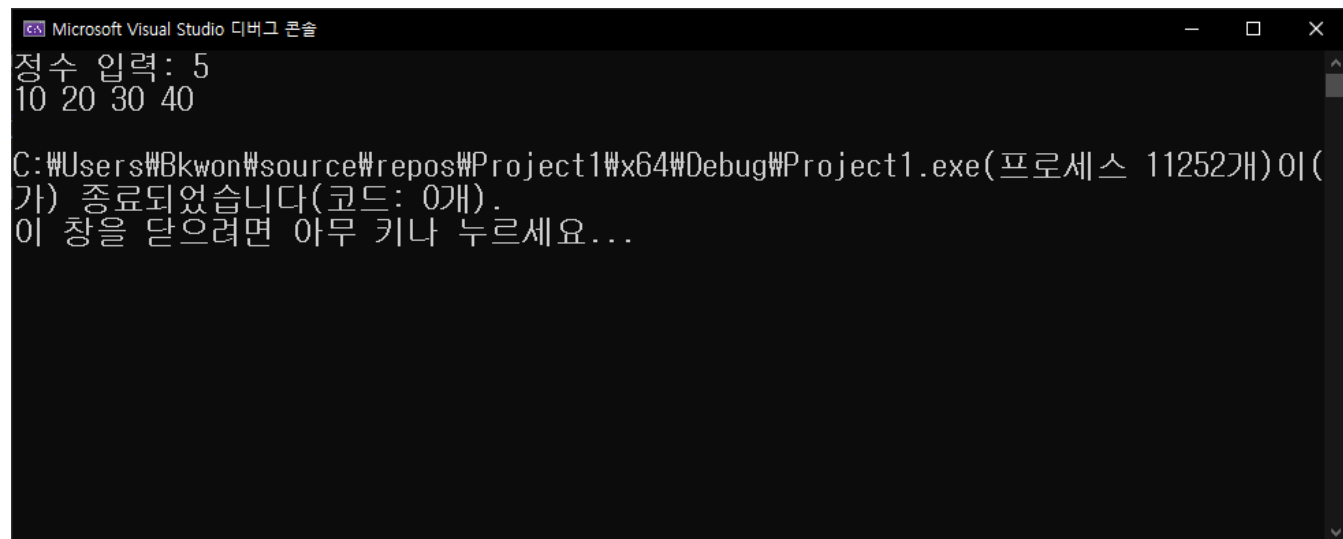
❖ 연습 문제 1. 정답 및 해설

```
1  /* example1.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int arr[5];
8      int total, j;
9
10     for (j = 0; j < 5; j++)
11     {
12         printf("%d층에 사는 사람의 수: ", j + 1);
13         scanf("%d", &arr[j]);
14     }
15
16     total = arr[0] + arr[1] + arr[2] + arr[3] + arr[4];
17     printf("현재 거주하는 총 인원: %d\n", total);
18     return 0;
19 }
```

03. 연습 문제

❖ 연습 문제 2.

- 길이가 9인 int형 정수 배열을 선언하고 나서 사용자로부터 숫자를 하나 입력 받습니다.
그리고 입력 받은 숫자의 구구단 계산 결과를 미리 선언해 놓은 배열에 저장합니다.
마지막으로 배열의 요소 중에서 짝수 번째(두 번째, 네 번째, ...) 위치에 존재하는 요소들을 출력하는 프로그램을 작성해 보세요.
(참고로 짝수 번째 위치에 존재하는 배열의 요소를 출력하려면 홀수를 인덱스 값으로 사용해야 할 것입니다.)



```
Microsoft Visual Studio 디버그 콘솔
정수 입력: 5
10 20 30 40

C:\Users\Bkwon\source\repos\Project1\Debug\Project1.exe(프로세스 11252개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

03. 연습 문제

❖ 연습 문제 2. 정답 및 해설

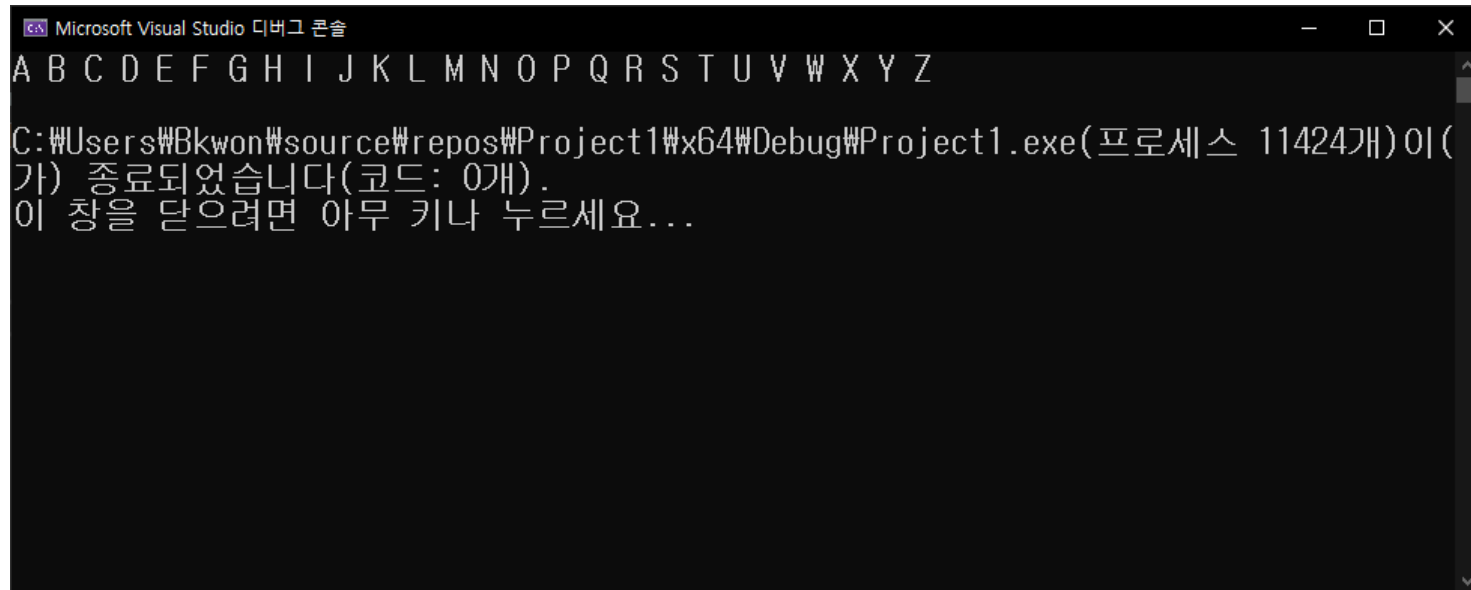
```
1  /* example2.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int arr[9];
8      int n, j;
9
10     printf("정수 입력: ");
11     scanf("%d", &n);
12
13     for (j = 1; j < 10; j++)
14     {
15         arr[j - 1] = n * j;
16     }
```

```
17     for (j = 0; j < 9; j++)
18     {
19         if (j % 2 == 1)
20             printf("%d ", arr[j]);
21     }
22     printf("\n");
23     return 0;
24 }
```

03. 연습 문제

❖ 연습 문제 3.

- 길이가 26인 char형 배열을 선언해서 문자 'A'부터 'Z'까지 저장합니다.
선언과 동시에 초기화하라는 뜻이 아닙니다. 반복문을 이용해서 저장하는 형태를 취해야 합니다.
그리고 배열에 모든 요소를 처음부터 끝까지 출력하는 프로그램을 작성해 보세요.
(아스키 코드 표를 참조해야만 문제를 해결할 수 있을까요?
아스키 코드 값을 모르고도 해결할 수 있는 방법은 없을까요?)



```
Microsoft Visual Studio 디버그 콘솔
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
C:\Users\Bkwon\source\repos\Project1\x64\Debug\Project1.exe(프로세스 11424개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

03. 연습 문제

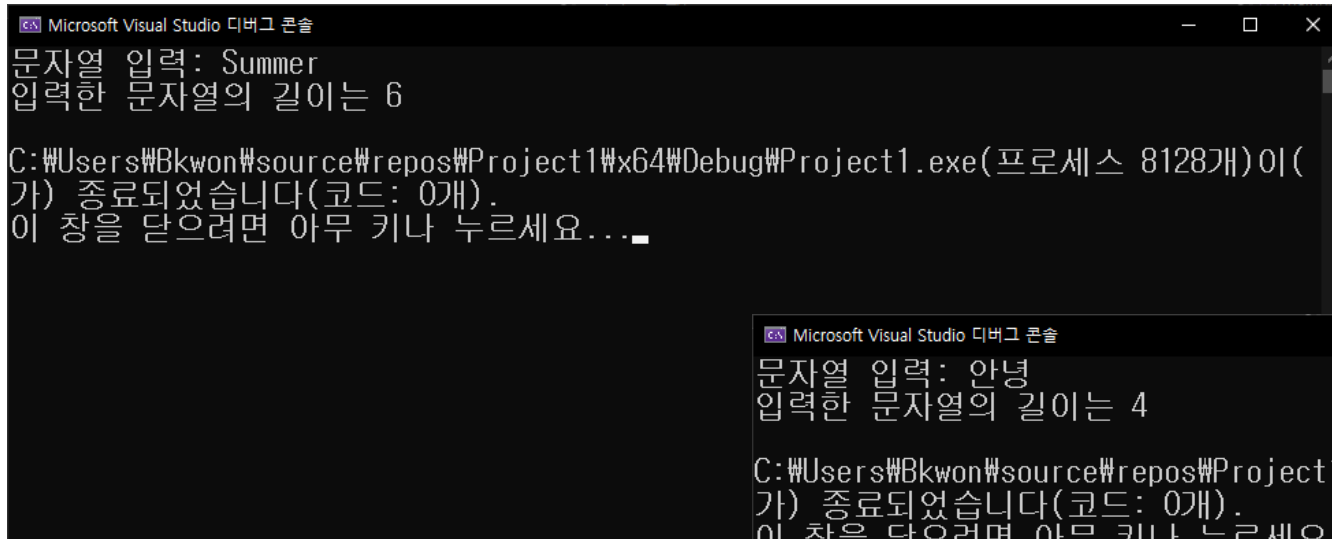
❖ 연습 문제 3. 정답 및 해설

```
1  /* example3.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      char arr[26];
8      char ch;
9      int j;
10
11     for (j = 0, ch = 'A'; j < 26; j++)
12         arr[j] = ch++;
13
14     for (j = 0; j < 26; j++)
15         printf("%c ", arr[j]);
16
17     printf("\n");
18     return 0;
19 }
```

03. 연습 문제

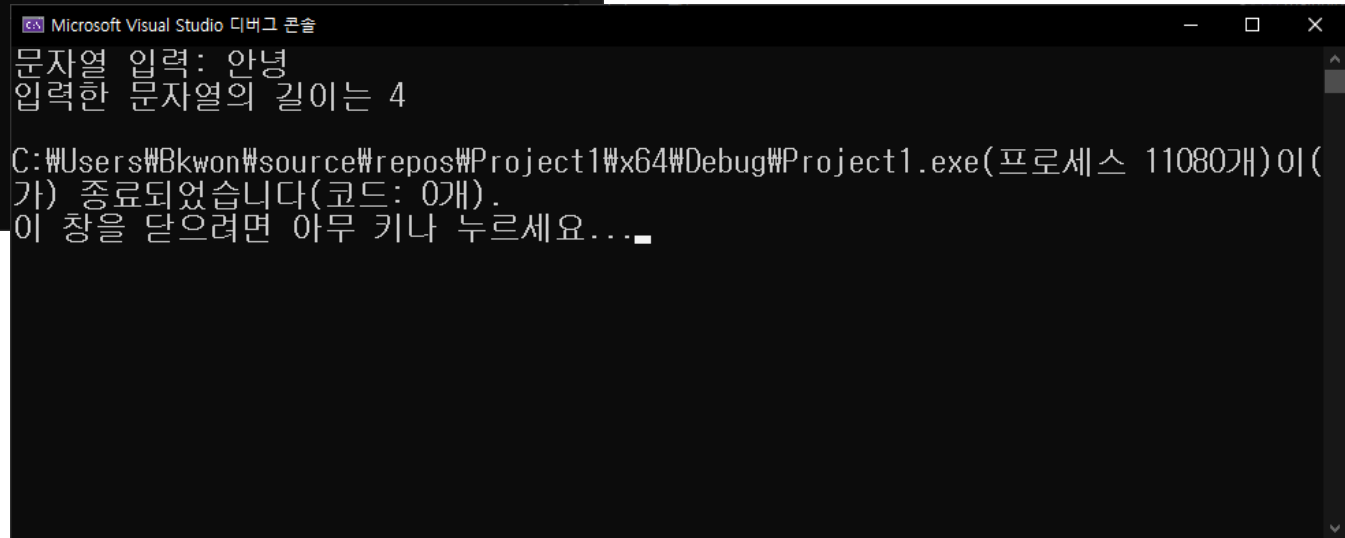
❖ 연습 문제 4.

- 사용자로부터 하나의 문자열을 입력받아서 입력받은 문자열의 길이를 출력하는 프로그램을 작성해 보세요. 널(Null) 문자는 문자열의 길이에서 제외시키는 것으로 가정합니다.



```
Microsoft Visual Studio 디버그 콘솔
문자열 입력: Summer
입력한 문자열의 길이는 6

C:\Users\Bkwon\source\repos\Project1\Debug\Project1.exe(프로세스 8128개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```



```
Microsoft Visual Studio 디버그 콘솔
문자열 입력: 안녕
입력한 문자열의 길이는 4

C:\Users\Bkwon\source\repos\Project1\Debug\Project1.exe(프로세스 11080개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```


03. 연습 문제

❖ 연습 문제 4. 정답 및 해설

```
1  /* example4.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      char string[100];
8      int len = 0;
9
10     printf("문자열 입력: ");
11     scanf("%s", string);
12
13     while (string[len] != '\0')
14         len++;
15
16     printf("입력한 문자열의 길이는 %d\n", len);
17     return 0;
18 }
```

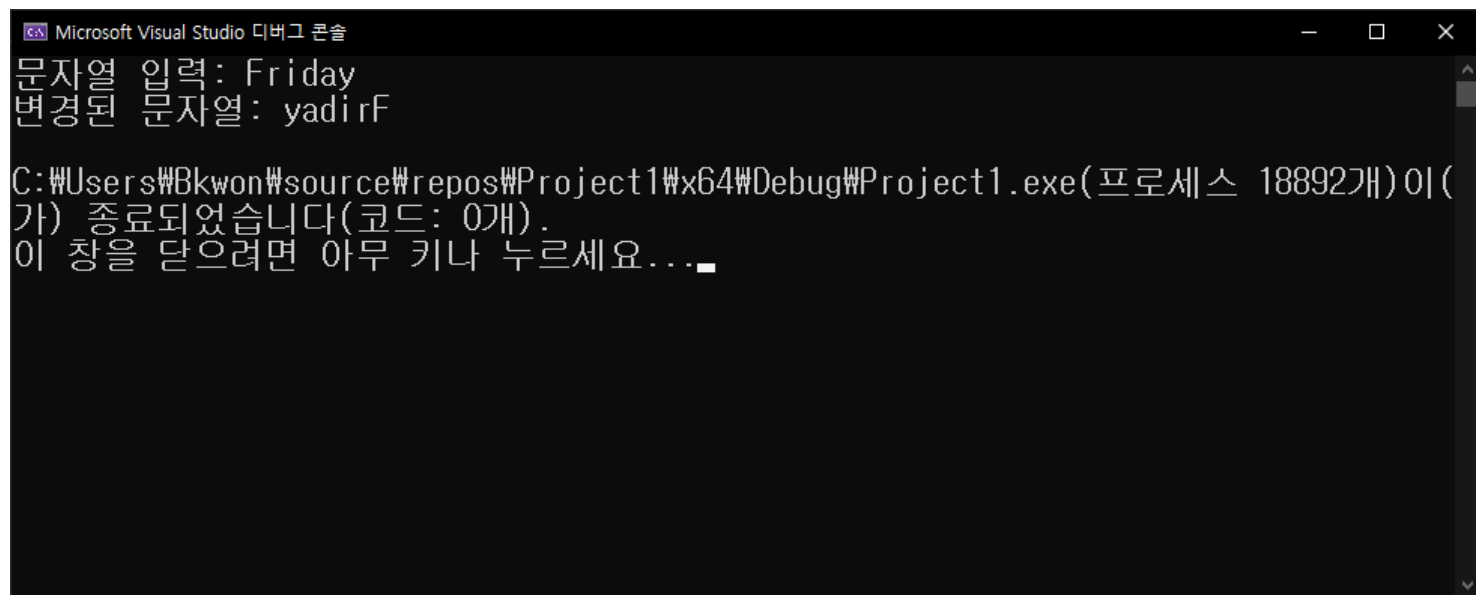
- ✓ 13번째 줄은 다음과 같이 표현해도 됩니다.
while (string[len] != 0)
- ✓ 문자는 변수에 저장될 때 아스키 코드 값으로 저장됩니다.
- ✓ 마찬가지로 문자는 비교 연산 시 아스키 코드 값으로 비교 연산을 합니다.
- ✓ 'W0'은 정수 0으로 변환되어 비교 연산을 수행하게 됩니다.

- ✓ 한글을 입력하면 실제 길이의 두 배가 계산됩니다.
- ✓ 컴퓨터는 한글을 표현하기 위해서 한글 한자 당 2바이트를 사용합니다.
- ✓ 그래서 실제 길이의 두 배로 계산되는 것입니다.

03. 연습 문제

❖ 연습 문제 5.

- 사용자가 입력한 문자열을 배열에 저장한 다음, 배열 내에서 문자열을 역순으로 뒤집습니다.
그리고 나서 제대로 뒤집어졌는지 확인하기 위해서 문자열을 출력하는 프로그램을 작성해 보세요.



```
Microsoft Visual Studio 디버그 콘솔
문자열 입력: Friday
변경된 문자열: yadirF

C:\Users\Bkwon\source\repos\Project1\x64\Debug\Project1.exe(프로세스 18892개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요..._
```

03. 연습 문제

❖ 연습 문제 5. 정답 및 해설

```
1  /* example5.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      char string[100];
8      int len = 0, j;
9
10     printf("문자열 입력: ");
11     scanf("%s", string);
12
13     while (string[len] != '\0')
14     {
15         len++;
16     }
```

```
17     for (j = 0; j < len/2; j++)
18     {
19         char temp = string[j];
20         string[j] = string[len - j - 1];
21         string[len - j - 1] = temp;
22     }
23
24     printf("변경된 문자열: %s\n", string);
25     return 0;
26 }
```

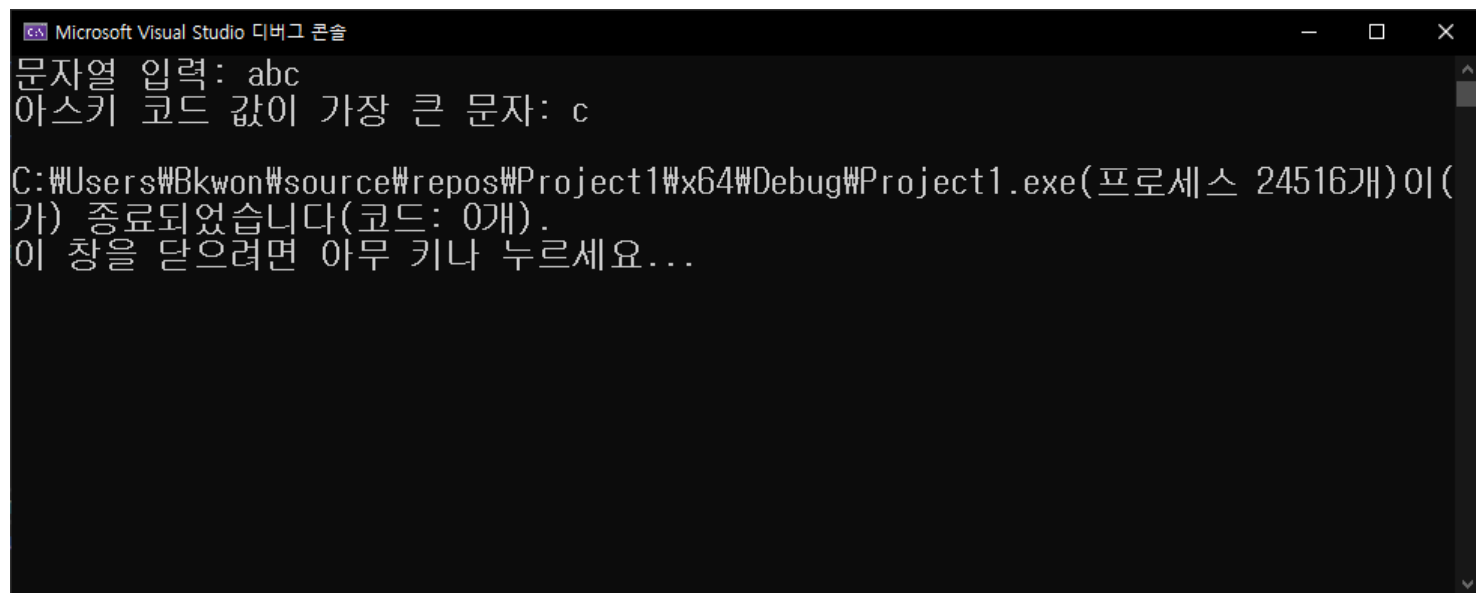
03. 연습 문제

❖ 연습 문제 6.

- 우선 문자열을 입력받습니다. 입력받은 문자열을 구성하고 있는 문자 중에서 아스키 코드 값이 가장 큰 문자를 찾아서 출력하는 프로그램을 구현해 보세요.

예를 들어 입력된 문자열이 "abc"라면 이중에서 아스키 코드 값이 가장 큰 문자는 'c'입니다.

따라서 c를 출력해 줍니다.



```
Microsoft Visual Studio 디버그 콘솔
문자열 입력: abc
아스키 코드 값이 가장 큰 문자: c

C:\Users\Bkwon\source\repos\Project1\x64\Debug\Project1.exe(프로세스 24516개)이(
가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

03. 연습 문제

❖ 연습 문제 6. 정답 및 해설

```
1  /* example6.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      char string[100];
8      int len = 0, j;
9      char big_ch = 0;
10
11     printf("문자열 입력: ");
12     scanf("%s", string);
13
14     for (j = 0; string[j] != '\0'; j++)
15     {
16         if (big_ch < string[j])
17             big_ch = string[j];
18     }
```

```
19     printf("아스키 코드 값이 가장 큰 문자: %c\n", big_ch);
20     return 0;
21 }
```

- ❖ 01. 배열의 이해와 배열의 선언 및 초기화 방법
- ❖ 02. 배열을 이용한 문자열 변수의 표현
- ❖ 03. 연습 문제

THANK YOU!

Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: bkwon@dongduk.ac.kr