

## 13 주차 Report

## [알림]

최근 SW 사업단 주관 프로그래밍 경진대회에 참가하였는데, 학교 Online Judge Platform 이 GNU GCC Compiler 로 채점한다고 하여 옛날에 사용하였던 Code::Block 을 setting 하여 잠시 이용했었습니다. 이때, 제 노트북에서 훨씬 가볍게 돌아가 Code::Block 을 쓰는 것이 더 효율적이라 생각되어 이번 레포트 풀이부터는 Visual Studio 2019 에서 Code::Block 으로 변경한다는 점을 알려드립니다.

문제 1) 정수 배열 A[]를 다른 정수 배열 B[]에 복사하는 함수를 작성하고 테스트하라.

```
void array_copy (int *A, int *B, int size){  
  
    int i;  
  
    for(i = 0; i < size; i++){  
        ...  
    }  
  
}
```

- 문제 분석 & 동작 설명

주어진 배열을 다른 빈 배열에 칸 맞춰서 넣으면 된다! 단, 문제에 주어진 사용자 정의 함수를 이용하자. 한번에 막 넣고 그러는 건 없으므로, 반복문을 돌리면서 셀 하나하나씩 다른 배열에 넣어주도록 해야 할 것이다. 근데, 출력 예시를 보았을 때 문제의 조건과는 좀 안 맞는 부분이 있다고 생각한다. B[]에 복사해야 하는데, A[]를 또 출력했기 때문이다. 이를 감안하여 코드를 짜도록 하자. A[]는 정해진 값이고, 이를 반복문을 이용하여 출력 예시에 맞게 출력해주자. 그리고 앞서 사용자 정의 함수를 돌리고 나온 B[]를 출력하면서 프로그램을 마무리하자.

- 프로그램 소스 (주석 포함)

```

main.c [Week13 Report 1] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> array_copy(int* A, int* B, int size) : void
main.c x
1 // 배열 Copy 프로그램
2 #include <stdio.h> // Standard Input Output 함수 헤더파일 전처리기
3
4 void array_copy (int *A, int *B, int size){ // 문제에서 주어진 배열 복사 사용자 정의 함수 array_copy
5 // 메인함수의 배열 A, B를 정수형 포인터 매개변수 A, B로 받고,
6 // 온 배열의 크기인 10을 정수형 매개변수 size로 받음
7
8 int i; // 아래의 반복문의 조건에 사용 할 정수형 지역변수 i 선언
9
10 for(i = 0; i < size; i++){ // 배열의 요소를 하나씩 각각 복사하기 위해 반복문 이용
11 B[i]=A[i]; // 각 Cell 마다 배열 A 에서 B로 복사
12 printf("%d ", B[i]); // 배열 B에 복사한 결과를 Cell 단위로 띄어쓰기 구분하여 출력
13 }
14 }
15
16
17
18 int main(void){ // main function
19
20 int A[10] = {1, 2, 3, 0, 0, 0, 0, 0, 0, 0}; // 크기가 10인 정수형 배열 A를 문제에 주어진 결과에 맞춰 초기화
21 int B[10]; // 배열 A로부터 복사할 배열 B 선언
22
23 int i; // 아래의 반복문의 조건에 사용 할 정수형 지역변수 i 선언
24
25 // 배열 A 출력
26 printf("A[] = "); // A[] =
27 for(i = 0; i < 10; i++) // 배열 A를 Cell 단위로 띄어쓰기 구분하여 출력하기 위해 반복문 이용
28 printf("%d ", A[i]); // 1 2 3 0 0 0 0 0 0 0
29
30 // 배열 B 출력
31 printf("\nB[] = "); // B[] =
32 array_copy(A, B, 10); // 배열 복사 사용자 정의 함수 array_copy 호출
33 // 배열 A, B와 배열의 크기 10을 매개변수로 올림
34
35 }
36

```

```

"D:\학교\대학교\충북대학교\1학년\기초컴퓨터프로그래밍 10분반\13주차\
A[] = 1 2 3 0 0 0 0 0 0 0
B[] = 1 2 3 0 0 0 0 0 0 0
Process returned 0 (0x0)    execution time : 0.047 s
Press any key to continue.

```

## 문제 2)

### 2-1. 다음 빈칸을 채우시오

- 컴퓨터에서 문자는 ASCII 코드로 나타낸다.
- 문자를 표현하는 자료형은 char 형이다.
- 문자열은 연속된 문자들의 모임으로 문자열의 끝은 NULL 문자로 표시한다.
- 문자열을 저장할 때는 char 형으로 선언된 배열에 저장한다.
- A, 'A', "A"의 차이점을 설명하는 다음 표의 빈칸을 채우시오

대상	설명
A	변수의 이름 A
'A'	문자 A
"A"	문자 A 로 이루어진 문자열

### 2-2. 다음 두 문장의 차이점을 설명하라

```
char a[] = "Hello World!"; // "Hello World!" 를 문자열 a 에 저장
char *p = "Hello World!"; // 문자열 상수는 메모리 영역 중 텍스트 세그먼트에 저장
// -> 이는 읽기만 가능하고, 값의 변경은 불가
```

### 2-3. 다음 프로그램에 오류가 있는지를 말하고, 있다면 수정하라.

Before

```
int main(void)
{
    char *p;
    scanf("%s", p);
}
```

After

```
int main(void) {
    char p[100];
    scanf("%s", p);
}
```

이유 : 문자열 포인터에 입력 값을 받기 위해서는 그 포인터가 초기화 되어있어야 한다.

2-4. 다음의 변수들에게 몇 바이트의 메모리가 할당되는가?

char = 1 바이트!!!!

```
(a) char str[] = "abc"; // a b c NULL > 4 바이트
(b) char *pc = "abc"; // 포인터의 Setting (in 32bit) > 4 바이트
(c) char str[10] = "abc";
    // a b c NULL NULL NULL NULL NULL NULL > 10 바이트
(d) char str[2][10] = {"abc", "del"}; // Total : 20 바이트
    // a b c NULL NULL NULL NULL NULL NULL NULL > 10 바이트
    // d e l NULL NULL NULL NULL NULL NULL NULL > 10 바이트
```

문제 3) 문자열을 입력으로 받아서 문자열에 포함된 모든 공백 문자를 삭제하는 함수를 작성하고 테스트하라.

Hint : 공백인지 검사를 한 후, 배열을 복사하는 코드

```
if(s[i] != ' ')
    tmp[k++] = s[i];
```

- 문제 분석 & 동작 설명

입력 받은 문자열에 공백을 확인하기 위해서는 반복문과 조건문을 사용해야 할 것이다. 반복문을 통해서 입력받은 문자열의 각 cell 을 조사하는데, 그 안에 조건문을 사용해서 그 cell 이 공백으로 가득 차 있는지 아니면, 문자로 차 있는지 구별해야 할 것이다. 그러한 과정으로, 문자로 차있는 cell 의 그 문자만 공백문자를 빼고 출력할 문자열에 차곡차곡 넣어 주도록 하자. 이때 반복문의 턴 수(입력받은 문자열의 Cell 번호 수)와 공백문자를 빼고 출력할 문자열의 Cell 번호 수가 같이 증가되도록 후자의 문자열의 cell 번호수를 제시된 힌트와 같이 k++로 구성해주도록 하자. 문자만 옮기는 과정이 끝난 후, 공백문자만을 뺀 문자열을 출력하고 프로그램을 종료하도록 하자.

- 프로그램 소스 (주석 포함)

```

main.c [Week13 Report 3] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
<global> main(void) : int
Program.c x main.c x
1 // 빈칸 빼주는 프로그램
2 #include <stdio.h> // Standard Input Output 함수 헤더파일 전처리기
3 #include <string.h> // strlen(문자열) 함수를 사용하기 위한 헤더파일 전처리기
4
5 int main(void) // main function
6 {
7     char input[100]; // 공백 문자가 있는 문자열을 입력 받을 크기가 100인 문자열 변수 선언
8     char output[100]; // 공백 문자를 제거할 문자열을 출력 할 크기가 100인 문자열 변수 선언
9
10    int i; // 입력 받은 문자열의 cell 하나씩의 구성을 판별하기 위한
11           // 반복문의 조건을 구성하는 정수형 변수 i 선언
12    int k = 0; // 반복문 내에서 같이 증가되는 변수 k를 0으로 초기화
13
14    printf("공백 문자가 있는 문자열을 입력하십시오 : "); // 사용자에게 문자열을 입력해라는 메시지 출력
15    gets(input); // 입력받은 문자열을 input 에 저장
16
17    for(i = 0; i < (int)strlen(input); i++){ // i가 0부터 1씩 증가하여 문자열 input의 크기 수가 될 때 까지 반복하는 반복문
18        if(input[i] != ' ') // i 번째 cell에 들어가는 문자가 공백이 아니라면
19            output[k++] = input[i]; // 그 문자는 output 이라는 배열의 같은 cell에 넣는다
20    }
21
22    printf("%s", output); // 문자열 output 출력
23
24    return 0; // main function 이 0으로 반환
25 }
26
27

```

```

"D:\학교\대학교\충북대학교\1학년\기초컴퓨터프로그래밍 10분반\13주차\
공백 문자가 있는 문자열을 입력하십시오 : i am a boy
iamaboy
Process returned 0 (0x0)   execution time : 10.449 s
Press any key to continue.

```

문제 4) 문자열을 사용자로부터 받아서 영어의 대문자는 소문자로, 소문자는 대문자로 변환하여 출력하는 프로그램을 작성하여 보자. `getchar()`를 이용하여 키보드로부터 문자를 하나 입력받는다. 입력받은 문자가 소문자이면 대문자로, 대문자라면 소문자로 변환하여 `putchar()`를 이용하여 출력한다. 만약 영어 알파벳의 범위를 벗어나는 문자가 들어오면 경고 메시지를 출력한다. 이 작업을 사용자가 '.'를 입력할 때까지 되풀이하는 프로그램을 작성한다.

- 문제 분석 & 동작 설명

대소문자를 변환하는 프로그램을 짜면 된다. `scanf()`와 `printf()`를 쓰는 것이 아닌, `putchar()`와 `getchar()`을 이용하여 알파벳을 입력받아야 한다. `putchar()`와 `getchar()`로 저장 및 출력하려는 변수를 정수형 변수로 지정을 해줘야 할 것이다. `while` 문을 이용하여 무한루프에 빠지도록 하자. 변수에 저장된 값들은 ASCII CODE 에 해당되는 2 진수 값일 것이다. ASCII CODE 표를 보면 대문자 값과 소문자 값의 차이가 32 이다. 따라서 조건문을 이용하여 입력된 알파벳의 ASCII CODE 값이 대문자 범위에 들어있으면 그 값에서 32 를 더해주어 해당되는 소문자를 `putchar()`로 출력해주도록 하고 입력된 알파벳의 ASCII CODE 값이 소문자 범위에 들어있으면 그 값에서 32 를 빼주어 해당되는 대문자를 `putchar()`로 출력해주도록 하자. 그리고 문자 '.'를 입력받았을 때는 `break` 문에 의해 무한루프를 탈출하여 프로그램을 종료시키도록 하고, 앞서 언급한 문자 외의 문자를 입력하였을 때는 다시 제대로 입력해달라고 메시지를 출력하자.

- 프로그램 소스 (주석 포함)

```

main.c [Week13 Report 4] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
<global> main(void) : int
main.c
1 // 대소문자 변환기
2 #include <stdio.h> // STanDard Input Output 함수 헤더파일 처리하기
3
4 int main(void) // main function
5 {
6     int input, output; // getchar로 입력받은 문자를 저장할 정수형 변수 input 과
7                       // 문자 변환 후 저장하여 putchar로 출력할 변수 output 선언
8
9     while(1) // 무한 반복
10    {
11
12        printf("문자를 입력하십시오 : "); // 사용자로부터 알파벳을 입력해라는 메시지 출력
13        input = getchar(); // 하나의 알파벳을 입력 받아 그 값을 정수형으로 변환하여 input에 저장
14
15
16        if (input >= 97 && input <= 122){ // 만약 입력 받은 알파벳이 소문자(a~z -> ASCII CODE : 97 ~ 122) 이라면
17            output = input - 32; // 그 알파벳에 해당하는 ASCII CODE에서 32를 빼줌 (A~Z-> ASCII CODE : 65 ~ 90)
18            putchar(output); // ASCII CODE의 뺄셈 연산을 한 값에 해당하는 문자를 출력
19            putchar('\n'); // 줄 띄어쓰기용
20        }
21
22        else if (input >= 65 && input <= 90){ // 만약 입력 받은 알파벳이 소문자(A~Z-> ASCII CODE : 65 ~ 90) 이라면
23            output = input + 32; // 그 알파벳에 해당하는 ASCII CODE에서 32를 더해주 (a~z -> ASCII CODE : 97 ~ 122)
24            putchar(output); // ASCII CODE의 덧셈 연산을 한 값에 해당하는 문자를 출력
25            putchar('\n'); // 줄 띄어쓰기용
26        }
27
28        else if (input == 46){ // 만약 입력 받은 문자가 . (ASCII CODE : 46) 이라면
29            break; // while 루프 탈출하여 프로그램 종료
30        }
31
32        else // 알파벳 대, 소문자가 아닌 다른 문자를 입력했다면
33            printf("\n올바른 알파벳을 입력하십시오.\n\n"); // 사용자에게 잘 입력해라고 메시지 출력하여 처음으로 되돌아가
34
35        getchar(); // 문제의 Hint에 나와있는 것 처럼 올바른 문자를 제거하기 위하여 getchar 재호출
36
37
38        return 0; // main function 의 0으로 반환
39    }
40

```

```

D:\학교\대학교\충북대학교\1학년\기초컴퓨터프로그래밍 10분반\13주차\
문자를 입력하십시오 : a
A
문자를 입력하십시오 : B
b
문자를 입력하십시오 : 7

올바른 알파벳을 입력하십시오.

문자를 입력하십시오 : .

Process returned 0 (0x0)   execution time : 8.053 s
Press any key to continue.

```