

2100998 이수찬 c++ 과제 21-3

● 과제1

```
// ****
// 제 목 : 문자열 연결
// 날 짜 : 2025년 3월 12일
// 작성자 : 2100998 이수찬
// *****

#define _CRT_SECURE_NO_WARNINGS //보안오류방지
#pragma warning(disable:6031) //리턴값관련 경고 방지
#include <stdio.h> //printf,scanf함수선언이 들어있는 stdio.h파일포함
#include<string.h> // strcmp 함수 사용하기 위한 헤더파일
int main(void) //메인함수의 머리
{//함수시작
    char str[100]; // 입력 받을 문자열 (최대 99자 + 널 문자)
    char str2[100] = ""; // 추가 입력 문자열 저장받기 위한 변수 선언
    while (1) { // 문자열을 입력하거나 그만두는 행위를 반복
        fputs("문자열 입력: ", stdout); // 문자열 입력 안내
        fgets(str, sizeof(str), stdin); // 문자열 입력 받기

        str[strcspn(str, "\n")] = '\0'; // 문자열에서 엔터 발견시 널문자로 변경

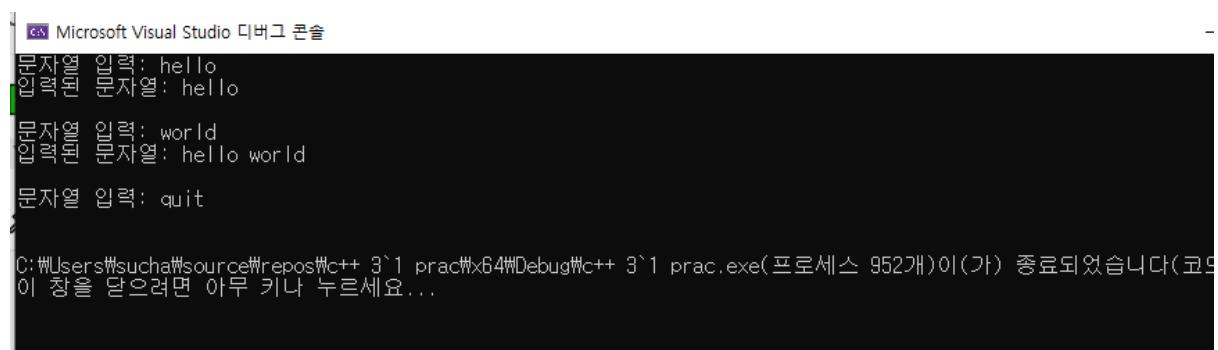
        if (strcmp(str, "quit") == 0) break; // str과 비교하여 입력된 값이 quit 일 경우 중단

        if (strlen(str2) > 0) strcat(str2, " "); // 추가 입력 받은 문자열이 0보다 클 때 다음 문자열 구분 위해 널문자 추가
        strcat(str2, str); // 문자열끼리의 합

        fputs("입력된 문자열: ", stdout); // 입력된 문자열 출력 위치 알리기
        puts(str2); // 입력된 문자열 출력
        putchar('\n'); // 줄 변경

    }// 반복문 종료

    putchar('\n'); // 출력 후 개행
}// 메인함수 종료
```



The screenshot shows the Microsoft Visual Studio Debug Console window. It displays the following text output:

```
Microsoft Visual Studio 디버그 콘솔
문자열 입력: hello
입력된 문자열: hello

문자열 입력: world
입력된 문자열: hello world

문자열 입력: quit

C:\Users\suchan\source\repos\c++_3\1_prac\x64\Debug\c++_3\1_prac.exe(프로세스 952개)이(가) 종료되었습니다(코드 0) 창을 닫으려면 아무 키나 누르세요...
```

● 과제2

```
// ****
// 제 목 : 엔터 널문자로 변경
// 날 짜 : 2025년 3월 12일
// 작성자 : 2100998 이수찬
// ****
#define _CRT_SECURE_NO_WARNINGS //보안오류방지
#pragma warning(disable:6031) //리턴값관련 경고 방지
#include <stdio.h> //printf,scanf함수선언이 들어있는 stdio.h파일포함
#include<string.h> // strcmp, strlen 함수 사용하기 위한 헤더파일

void RemoveBSN(char str[]); //엔터 널문자 변환 함수 선언
int my_strlen(char str[]); //my_strlen 함수선언

int main(void) //메인함수의 머리
{//함수시작
    char str[100]; //적을 문자열 변수 선언
    printf("문자열 입력: "); // 문자열 위치 입력구간 출력
    fgets(str, sizeof(str), stdin); //문자열 입력 위한 함수
    printf("길이: %d, 내용: %s \n", my_strlen(str), str); //my_strlen함수에서
    가져온 길이와 문자 출력
    RemoveBSN(str); //BSN 함수 작동시키기 위한 코드
    printf("길이: %d, 내용: %s \n", my_strlen(str), str); // BSN 작동한 후 길이와
    문자 출력
    return 0; //함수 종료 후 반환
}

// 메인함수 종료

int my_strlen(char str[]) { // 선언한 함수 정의
    int j = 0; // 길이 구하기 위해 선언한 함수 초기화
    while(str[j]!='\0'){ //널문자 아니면 작동
        j++; //널문자 아닐 시 카운트
    } //반복문 종단
    return j; // j로 길이 반환
} // 함수 정의 종료

void RemoveBSN(char str[]) { //엔터 널문자로 변경하기 위한 함수 정의
    int len = my_strlen(str); // 문자열의 길이 확인
    if (len > 0 && str[len - 1] == '\n') { // 마지막 문자가 '\n'이면
        str[len - 1] = '\0'; // 개행 문자를 널 문자로 변경
    } // 조건문 종료
} // 함수 정의 종료
```

```
-----  
문자열 입력: hello world  
길이: 12, 내용: hello world  
  
길이: 11, 내용: hello world  
C:\Users\sucha\source\repos\C++ 3\1 prac\x64\Debug\C++ 3\1 prac.exe(프로세스 17244)  
이 창을 닫으려면 아무 키나 누르세요...  
C:
```

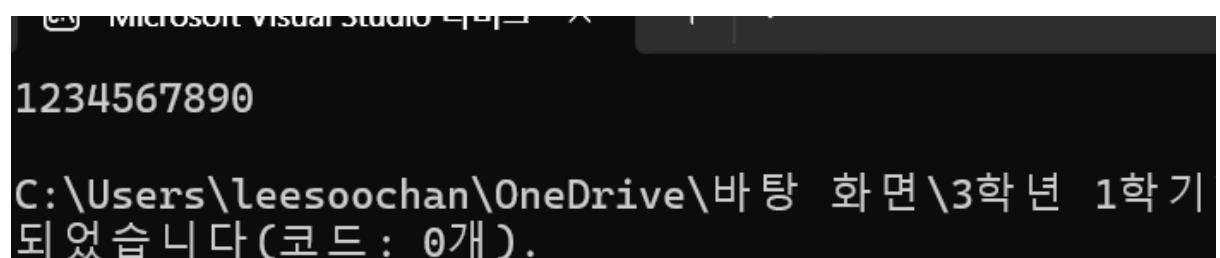
● 과제3

```
// ****
// 제 목 : 문자열 복사
// 날 짜 : 2025년 3월 12일
// 작성자 : 2100998 이수찬
// ****
#define _CRT_SECURE_NO_WARNINGS //보안오류방지
#pragma warning(disable:6031) //리턴값관련 경고 방지
#include <stdio.h> //printf,scanf함수선언이 들어있는 stdio.h파일포함
#include<string.h> // strcmp 함수 사용하기 위한 헤더파일

void my_strcpy(char str2[],char str1[]); //함수 정의 위해 선언

int main(void) //메인함수의 머리
{//함수시작
    char str1[20] = "1234567890"; //복사할 문자열 선언
    char str2[20] = ""; //복사 받을 문자열 선언
    my_strcpy(str2, str1); // 정의한 함수를 이용하여 복사하기 위한 함수
    puts(str2); //복사 받은 함수 출력
    return 0; //함수 반환 후 종료
}// 메인함수 종료

void my_strcpy(char str2[],char str1[]) { //my_strcpy 정의 위한 함수
    int j = 0; //복사할 문자열의 문자 위치 알리기 위한 정수선언
    while (str1[j] != '\0') { //복사 할 문자열이 널인지 아닌지 확인 후 아닐 경우
        작동
            str2[j] = str1[j]; //복사 할 문자열을 받을 문자열에 대입.
            j++; //위치 증가
    } //반복문 종료
    str1[j] = '\0'; //널문자 대입하여 종료시키기
}
```



1234567890

C:\Users\leesoochan\OneDrive\바탕 화면\3학년 1학기 되었습니다(코드 : 0개).

● 과제4

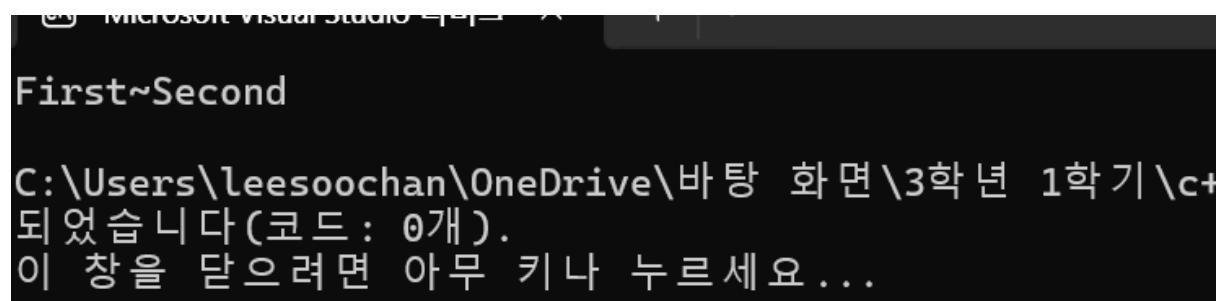
```
// ****
// 제 목 : 문자열 붙이기
// 날 짜 : 2025년 3월 12일
// 작성자 : 2100998 이수찬
// ****
#define _CRT_SECURE_NO_WARNINGS //보안오류방지
#pragma warning(disable:6031) //리턴값관련 경고 방지
#include <stdio.h> //printf,scanf함수선언이 들어있는 stdio.h파일포함
#include<string.h> // strcmp 함수 사용하기 위한 헤더파일

void my_strcat(char str1[], char str2[],char result[]); //문자열 붙이기 위한 함수 선언

int main(void) //메인함수의 머리
{//함수시작
    char str1[20] = "First~"; // 복사할 문자열 초기화
    char str2[20] = "Second"; // 복사할 문자열 초기화
    char result[100] = ""; // 결과 출력할 문자열 초기화
    my_strcat(str1, str2,result); //선언한 함수 실행
    puts(result); // 결과 문자열 출력
    return 0; //함수 반환 후 종료
}// 메인함수 종료

void my_strcat(char str1[],char str2[],char result[]) { //my strcpy 정의 위한 함수
    int j = 0,i=0; //복사할 문자열의 문자 위치 알리기 위한 정수선언
    while (str1[j] != '\0') { //복사 할 문자열이 널인지 아닌지 확인 후 아닐 경우
        작동
        result[j]= str1[j]; // 복사 할 문자열을 결과 문자열에 대입 .
        j++; //위치 증가
    } //반복문 종료

    while (str2[i] != '\0') { //복사 할 문자열이 널인지 아닌지 확인 후 아닐 경우
        작동
        result[j] = str2[i]; // 결과 문자열은 이어서 받고 새로 복사해줄
        문자열은 처음부터 카운트하며 문자열 추가
        j++; // 결과문자열 인덱스 증가
        i++; //복사 할 문자열 인덱스 증가
    } // 반복문 증가
    str2[j] = '\0'; //널문자 대입하여 종료시키기
}
```



The screenshot shows the Microsoft Visual Studio interface with the output window open. The window displays the following text:

```
First~Second

C:\Users\leesoochan\OneDrive\바탕 화면\3학년 1학기\c+
되었습니다(코드 : 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

● 과제5

```
// ****
// 제 목 : 문자열 크기비교
// 날 짜 : 2025년 3월 13일
// 작성자 : 2100998 이수찬
// ****
#define _CRT_SECURE_NO_WARNINGS //보안오류방지
#pragma warning(disable:6031) //리턴값관련 경고 방지
#include <stdio.h> //printf,scanf함수선언이 들어있는 stdio.h파일포함
#include<string.h> // strcmp 함수 사용하기 위한 헤더파일

int main(void) //메인함수의 머리
{//함수시작
    char str[100]; // 문자열 선언
    char temp[100] = ""; // 빈칸 선언
    char temp1[100] = "zzzzzzzzzzzzzzzzzz"; // 가장 큰 문자열 선언
    for ( int i = 0; i < 5; i++) { // 5번 반복
        fputs("문자열 입력: ",stdout); // 문자열 입력 위치 확인
        fgets(str, sizeof(str), stdin); // 문자열 입력
        if (strcmp(str, temp) > 0) { // 입력된 문자열 중 가장 큰 수 확인
            strcpy(temp, str); // temp보다 str이 크다면 temp에 str 저장
        }// 조건문 종료
        if (strcmp(str, temp1) < 0) { // 입력된 문자열 중 가장 작은 수 확인
            strcpy(temp1, str); // temp보다 str이 작다면 temp에 str 저장
        }// 조건문 종료
    }// 반복문 종료
    fputs("제일 앞: ", stdout); //제일 작은 수 확인하기 위한 위치
    fputs(temp1, stdout); //제일 작은 수 출력
    putchar('Wn');// 줄 변경

    return 0; //함수 반환 후 종료
}// 메인함수 종료
```

```
문자열 입력 : hello
문자열 입력 : world
문자열 입력 : apple
문자열 입력 : melon
문자열 입력 : banana
제일 앞 : apple
제일 뒤 : world

C:\Users\leesoochan\OneDrive\바탕 화면\3학년 1학기\c언어\Project1\x64\Debug\Project1.exe(프로세스 18836개)
단계별 실행 중입니다(코드 : 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

- 21장 정리문제

- 1) 문자는 컴퓨터에서 어떻게 저장되는가?
 - ⇒ 아스키 코드의 정수값으로 저장
- 2) 문자입력함수를 아는대로 설명하라.
 - ⇒ Scanf(): 문자나 문자열을 입력받음
 - ⇒ Getchar(): 한 문자 입력
 - ⇒ Gets(): 문자열 입력
 - ⇒ Fgets(): 문자열 입력(버퍼 크기 지정 가능)
- 3) 문자출력함수를 아는대로 설명하라.
 - ⇒ Printf(): 문자 또는 문자열을 출력
 - ⇒ Putchar(): 문자 하나를 출력
 - ⇒ Puts(): 문자열 출력 후 줄 바꿈

- 4) 다음 문장을 실행할 때 문제점을 설명하라.

```
scanf("%d",&num);
scanf("%c",&ch);
```

- ⇒ 처음 scanf()에서 정수 입력한 후 엔터키가 입력 버퍼에 남는다.
⇒ 두 번째 scanf("%c",&ch);는 이 엔터를 문자로 받아들여 ch에 저장

- 5) 4번의 해결방법을 설명하라. 게시판의 문자 및 문자열 입력시 엔터키 처리방법을 참고할 것

- ⇒ 1. Getchar()로 버퍼 비우기

```
Scanf("%d", &num);
Getchar();
Scanf("%c", &ch);
```

- ⇒ 2. 공백 처리 활용

```
Scanf("%d",&num);
Scanf(" %c", &ch);
```

- 6) 널문자의 역할은?

- ⇒ C 언어 문자열의 끝을 나타내는 종료 문자이다.

- ⇒ 문자열의 끝을 명확히 하여 메모리 접근 오류를 방지합니다.

- 7) 널문자를 자동으로 저장해주는 경우를 설명하라

- ⇒ 문자열 리터럴로 초기화할 때

```
Char str[]="Hello"; // 마지막에 '₩0' 자동 추가
```

- ⇒ Gets()와 fgets()로 입력받을 때

```
Char str[20];
Gets(str); // '₩0'이 자동 추가됨.
```

- 8) scanf함수로 문자열을 입력받을 때 종료문자는?

- ⇒ '₩0'(빈 공간)에서 입력이 종료

- 9) gets함수로 문자열을 입력받을 때 종료문자는?

- ⇒ 엔터키('₩')를 누를 때 입력이 종료된다

- ⇒ 입력된 문자열의 끝에 널문자가 자동으로 추가된다.

10) scanf와 gets함수의 차이점은?

⇒

구분	Scanf()	Gets()
입력방식	공백에서 종료	엔터키까지 한 줄 전체 입력
공백 처리	공백 이후의 문자열 무시	공백 포함 전체 입력
안전성	입력 길이 초과시 버퍼 오버플로우 발생가능	버퍼 오버플로우 위험 있음(대신 fgets() 사용 권장)

11) 문자열 출력함수를 설명하라.

⇒

함수	설명	특징
Printf()	문자열을 원하는 형식으로 출력	포맷 지정자 사용(%s)
Puts()	문자열 출력 후 줄 바꿈(\n) 자동 삽입	문자열 전용, 줄 바꿈 추가
Fputs()	문자열 출력	줄바꿈 없음, 더 유연함
Putchar()	문자 하나 출력	한 문자씩 출력(반환값: 출력한 문자)

12) C표준라이브러리 함수인 strcpy, strlen, strcat과 똑같은 기능을 갖는 함수를 작성하시오. 함수를 활용하는 예제까지 포함하여 만들 것.

⇒

```
void my_strcpy(char* dest, const char* src) {
    while (*src != '\0') {
        *dest = *src;
        dest++;
        src++;
    }
    *dest = '\0'; // 널 문자 추가
}

// 2. strlen과 동일한 기능의 함수 (문자열 길이 구하기)
int my_strlen(const char* str) {
    int length = 0;
    while (*str != '\0') {
```

```

        length++;
        str++;
    }
    return length;
}

// 3. strcat과 동일한 기능의 함수 (문자열 연결)
void my_strcat(char* dest, const char* src) {
    // dest 끝을 찾음
    while (*dest != '\0') {
        dest++;
    }
    // src를 dest에 복사
    while (*src != '\0') {
        *dest = *src;
        dest++;
        src++;
    }
    *dest = '\0'; // 널 문자 추가
}

int main(void) {
    char str1[50] = "Hello, ";
    char str2[] = "World!";
    char copy[50];

    // my strcpy 사용 예제
    my strcpy(copy, str1);
    printf("복사된 문자열: %s\n", copy);

    // my strlen 사용 예제
    printf("str1의 길이: %d\n", my_strlen(str1));

    // my strcat 사용 예제
    my strcat(str1, str2);
    printf("문자열 연결 결과: %s\n", str1);

    return 0;
}

```

13) 문자열 저장시 아래 2문장 모두 가능합니다. 2가지방법의 차이를 설명하라.

```

char str[20] = "apple";
char *p = "pineapple";

```

구분	Char str[20] = "apple";	Char *p ="pineapple";
메모리 위치	**스택(stack)**에 저장됨	**읽기 전용 메모리(상수 영역)**에 저장됨
수정 가능 여부	수정 가능 (str[0] = 'A')	수정 불가능 (p[0] = 'P'; 오류 발생)

메모리 크기	선언한 크기(20바이트) 사용	문자열 길이에 따라 동 적 할당(8바이트)
사용 목적	문자열을 수정하거나 복사	상수 문자열을 참조할 때 사용

1. **char str[20] = "apple";**

- 스택 메모리에 저장되며, 배열의 크기(20바이트)가 고정됩니다.
- 문자열을 수정 가능합니다.

```
str[0] = 'A'; // 가능
printf("%s\n", str); // "Apple"
```

2. **char *p = "pineapple";**

- 문자열 리터럴 "pineapple"은 **읽기 전용 메모리(상수 영역)**에 저장됩니다.
- 포인터 p는 해당 주소를 가리키고 있으며, **수정하면 오류**가 발생합니다.

```
p[0] = 'P'; // 오류 발생 (읽기 전용 메모리 변경 시도)
printf("%s\n", p); // 가능: "pineapple"
```