

## <로봇 보고서>

4일차\_최우진

1.

### <메인함수>

```
int main(int argc, const char * argv[]) {
    struct string_input str1;
    struct string_input str2;
    int f;

    printf("문자열 str1을 입력하시오. ");
    scanf("%s",str1.str);
    printf("문자열 str2을 입력하시오. ");
    scanf("%s",str2.str);

    f=string_len(str1.str, str2.str);
    if(f==-1){
        printf("str1가 더 길니다.");
    }
    else if(f==1){
        printf("str2가 더 길니다.");
    }
    else{
        printf("문자열의 길이가 같습니다.");
    }

    return 0;
}
```

-> 구조체 변수 선언해주고 문자를 구조체를 사용해서 받았습니다. string\_len 함수 선언하고 return 값을 f에 저장한뒤 조건문 걸어서 출력했습니다.

### <string\_len함수>

```
int string_len(char str1[],char str2[]){
    int n1,n2;
    for (int i=0; i<100; i++) {
        if(str1[i]!='\0'){
            n1 = i;
            break;
        }
    }
    for (int i=0; i<100; i++) {
        if(str2[i]!='\0'){
            n2 = i;
            break;
        }
    }
    if(n1>n2){
        return -1;
    }
    else if(n1<n2){
        return 1;
    }
    else{
        return 0;
    }
}
```

-> 길이를 비교하기 위한 함수입니다. 두 배열의 길이를 비교하고 조건에 맞게 return 하도록 구현했습니다.

<구조체>

```
struct string_input{  
    char str[100];  
};
```

-> 문자열 입력받을때 사용하기 위해 정의했습니다.

출력 결과

```
문자열 str1을 입력하십시오. asdf  
문자열 str2을 입력하십시오. asgdge  
str2가 더 깁니다.Program ended with
```

```
문자열 str1을 입력하십시오. asdfgh  
문자열 str2을 입력하십시오. asd  
str1가 더 깁니다.Program ended with
```

```
문자열 str1을 입력하십시오. asdf  
문자열 str2을 입력하십시오. asdf  
문자열의 길이가 같습니다.Program end
```

2.

<메인 함수>

```
int main(int argc, const char * argv[]) {  
    struct string_input str1;  
    struct string_input str2;  
    printf("문자열을 입력하십시오(str1). ");  
    scanf("%s",str1.str);  
    printf("문자열을 입력하십시오(str2). ");  
    scanf("%s",str2.str);  
    string_cat(str1.str, str2.str);  
    return 0;  
}
```

-> 구조체 변수 선언해서 문자 입력 받고 string\_cat함수 호출했습니다.

<구조체>

```
struct string_input{  
    char str[100];  
};
```

-> 문자형 배열 사용하려고 구조체 정의해합니다.

## <string\_cat 함수>

```
char* string_cat(char s1[], char s2[]) {
    int n1, n2;
    char out[100];
    for (int i=0; i<100; i++) {
        if(s1[i]=='\0'){
            n1 = i;
            break;
        }
    }
    for (int i=0; i<100; i++) {
        if(s2[i]=='\0'){
            n2 = i;
            break;
        }
    }

    if(n1<n2){
        int i = 0;
        int j = 0;
        while (s2[j] != '\0') {
            out[i]=s2[j];
            i++;
            j++;
        }
        s1[i] = "\0";
        printf("복사되었습니다.\n");
        printf("%s\n", s1);
    }
    else{
        printf("복사할 수 없습니다.\n");
    }

    return 0;
}
```

-> 우선 배열의 길이를 구해줬습니다. 조건 걸어서  $n1 < n2$  일때만 복사해서 출력하게했고 아닐시는 복사할 수 없게 구현했습니다.

## 출력 결과

```
문자열을 입력하시오(str1). robit
문자열을 입력하시오(str2). good
복사할 수 없습니다.
Program ended with exit code: 0
```

```
문자열을 입력하시오(str1). good
문자열을 입력하시오(str2). robit
복사되었습니다.
good
Program ended with exit code: 0
```

3.

### <메인 함수>

```
int main(int argc, const char * argv[]) {
    struct string_input str1;
    struct string_input str2;
    int f;

    printf("Str1 = ");
    scanf("%s",str1.str);
    printf("Str2 = ");
    scanf("%s",str2.str);

    f=stringCompare(str1.str, str2.str);

    if (f==1) {
        printf("Str1가 더 빠릅니다.\n");
    }
    else if (f==-1) {
        printf("Str2가 더 빠릅니다.\n");
    }
    else{
        printf("Str1==Str2.\n");
    }

    return 0;
}
```

-> 구조체 변수를 선언해주고 입력 받은 후 함수 호출 받아서 리턴받는 값을 f에 넣고 조건에 맞게 출력되게 구현했습니다.

### <stringCompare 함수>

```
int stringCompare(const char* str1, const char* str2) {
    //아스키코드가 소문자가 더 크니깐 대문자 소문자 크기 바꿔줌
    while (*str1&&*str2) {
        char cc1 = ((*str1>='A') && (*str1<='Z')) ? (*str1) : (*str1);
        char cc2 = ((*str2>='A') && (*str2<='Z')) ? (*str2) : (*str2);

        if(cc1<cc2){
            return 1;
        }
        else if (cc1>cc2){
            return -1;
        }

        str1++;
        str2++;
    }
    return 0;
}
```

-> 위에 주석 무시해주세요..! 처음에는 대소문자 바꿔서 구현하려고 크기비교하려 했었는데 뭔가 자꾸 반대로 돼서 그냥 cc1,cc2에 각각 str1,str2를 저장했습니다. 그 후 비교해서 return해줬습니다. \*문자형배열변수는 첫번째를 배열을 의미하기에 반복문에 넣어서 쪽 비교할 수 있게 구현했습니다.

<구조체>

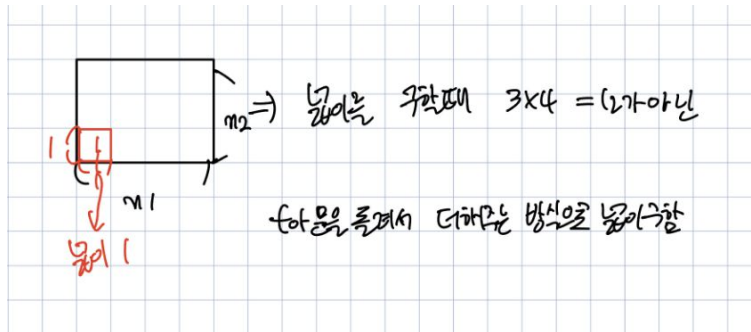
```
struct string_input{  
    char str[100];  
};
```

-> 문자형 배열 사용하려고 구조체 정의해줍니다.

4.

<알고리즘 설명>

넓이를 구할때 가로와 세로 길이를 구해서 곱하면서 구하는 방식이 아닌 좌표로 생각해서 1씩 더해서 구하는 방식을 사용했습니다.



이렇게 두 넓이를 따로 구해서 더해주고 그 후 4중for문을 써서 겹칠때마다 -1해주는 방식으로 구현했습니다.

<메인 함수>

너무 길어서 사진 첨가는 생략하겠습니다. 코드 확인 부탁드립니다. 비교값의 경우를 12가지로 나눠서 구현했습니다.

<<<< ~ >>>>까지 12개의 경우를 다 조건걸어서 구현했습니다

<구조체>

```
struct string_input{  
    char str[100];  
};
```

-> 문자형 배열 사용하려고 구조체 정의해줍니다.

출력 결과

```
1번 좌표 : -2 -3
2번 좌표 : 2 4
3번 좌표 : -1 -1
4번 좌표 : 3 3
교집합 빼기 전: 44
합집합 뺀 후 : 32
Program ended with exit code: 0
```