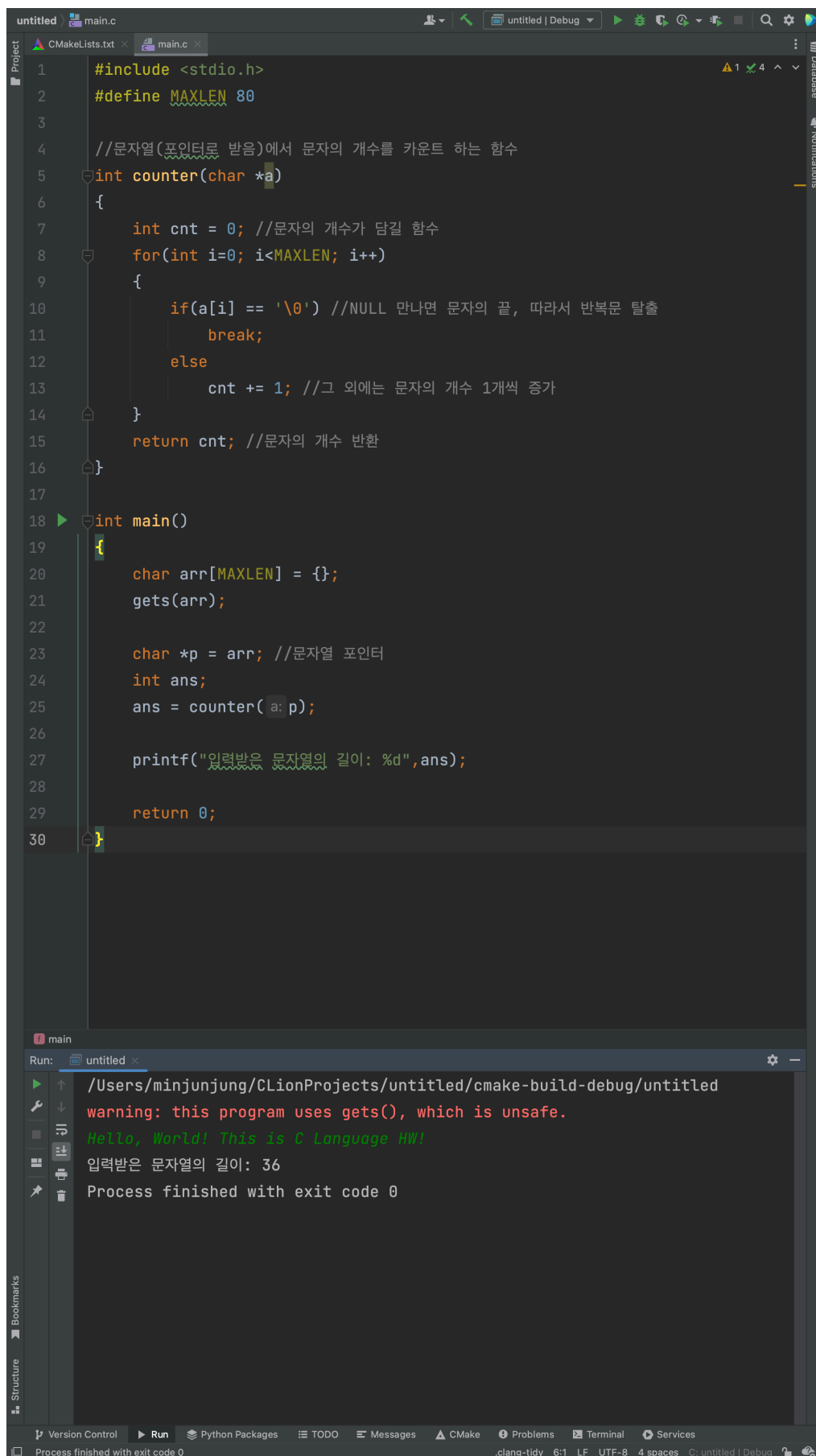

레포트 #3 (“포인터&구조체”)

SW융합코딩2(2분반 우진운 교수님)

32184074 컴퓨터공학과 정민준



1번 코드 & 결과



```
1 #include <stdio.h>
2 #define MAXLEN 80
3
4 //문자열(포인터로 받음)에서 문자의 개수를 카운트 하는 함수
5 int counter(char *a)
6 {
7     int cnt = 0; //문자의 개수가 담길 함수
8     for(int i=0; i<MAXLEN; i++)
9     {
10         if(a[i] == '\0') //NULL 만나면 문자의 끝, 따라서 반복문 탈출
11             break;
12         else
13             cnt += 1; //그 외에는 문자의 개수 1개씩 증가
14     }
15     return cnt; //문자의 개수 반환
16 }
17
18 int main()
19 {
20     char arr[MAXLEN] = {};
21     gets(arr);
22
23     char *p = arr; //문자열 포인터
24     int ans;
25     ans = counter(a: p);
26
27     printf("입력받은 문자열의 길이: %d",ans);
28
29     return 0;
30 }
```

Run: untitled

```
/Users/minjunjung/CLionProjects/untitled/cmake-build-debug/untitled
warning: this program uses gets(), which is unsafe.
Hello, World! This is C Language HW!
입력받은 문자열의 길이: 36
Process finished with exit code 0
```

1번 결과 & strlen 값과 비교

```
1 #include <stdio.h>
2 #include <string.h>
3 #define MAXLEN 80
4
5 //문자열(포인터로 받음)에서 문자의 개수를 카운트 하는 함수
6 int counter(char *a)
7 {
8     int cnt = 0; //문자의 개수가 담길 함수
9     for(int i=0; i<MAXLEN; i++)
10     {
11         if(a[i] == '\0') //NULL 만나면 문자의 끝, 따라서 반복문 탈출
12             break;
13         else
14             cnt += 1; //그 외에는 문자의 개수 1개씩 증가
15     }
16     return cnt; //문자의 개수 반환
17 }
18
19 int main()
20 {
21     char arr[MAXLEN] = {};
22     gets(arr);
23
24     char *p = arr; //문자열 포인터
25     int ans;
26     ans = counter(a: p);
27
28     //길이 비교를 위한 strlen 추가
29     printf("strlen을 이용한 문자열의 길이: %d\n", strlen(s: arr));
30     printf("입력받은 문자열의 길이: %d", ans);
31
32     return 0;
33 }
```

Run: untitled

```
/Users/minjungjung/CLionProjects/untitled/cmake-build-debug/untitled
warning: this program uses gets(), which is unsafe.
Hello, World! This is C Language HW!
strlen을 이용한 문자열의 길이: 36
입력받은 문자열의 길이: 36
Process finished with exit code 0
```

1번 설명

1. gets로 배열 arr에 문자열을 입력받습니다.
2. 테스트에 사용한 문자열은 “Hello, World! This is C Language HW!”입니다.
3. 포인터를 인자로 받는 counter 함수를 만듭니다.
4. 문자의 개수를 셀 때 NULL을 만나면 반복문을 탈출하고 그 외에는 cnt 변수를 1씩 증가시킵니다.
5. 포인터 변수 p에 문자열을 할당하고 counter 함수에 넣어서 ans에 cnt를 반환받습니다.
6. 정답 변수인 ans를 출력하고 프로그램을 종료합니다.
7. 위의 “1번 결과 및 비교”에서 보이는 것처럼 strlen을 이용한 값과 counter 함수를 이용한 값이 동일함을 볼 수 있습니다.

2번 코드-1

```
untitled main.c
CMakeLists.txt main.c
1 #include <math.h>
2 #include <stdio.h>
3
4 //좌표 구조체
5 struct point{
6     int x,y;
7 };
8
9 int main()
10 {
11     struct point p[5]; //좌표값을 담을 구조체 배열 선언
12     int ans_x1,ans_y1,ans_x2,ans_y2;
13     double ans;
14
15     for(int i=0; i<5; i++)
16     {
17         scanf("%d",&p[i].x); //구조체 배열에 x좌표 입력
18         scanf("%d",&p[i].y); //구조체 배열에 y좌표 입력
19     }
20
21     //2중 반복문으로 비교
22     for(int i=0; i<4; i++)
23     {
24         for(int j=i+1; j<5; j++)
25         {
26             if(i==0 && j==1)
27             {
28                 //두 좌표 간의 거리가 담길 변수 선언 및 계산 첫 시행 이므로 정답 변수에 할당
29                 ans = sqrt(pow(p[i].x-p[j].x,2) + pow(p[i].y-p[j].y,2));
30                 ans_x1 = p[i].x;
31                 ans_y1 = p[i].y;
32                 ans_x2 = p[j].x;
33                 ans_y2 = p[j].y;
34             }
35             else
36             {
37                 double tmp; //두 좌표 간의 거리가 담길 변수 선언 및 계산
38                 tmp = sqrt(pow(p[i].x-p[j].x,2) + pow(p[i].y-p[j].y,2));
39
40                 if(tmp<ans) //기존 값보다 작다면 최단 거리 및 좌표값 갱신
41                 {
42                     ans = tmp;
43                     ans_x1 = p[i].x;
44                     ans_y1 = p[i].y;
45                     ans_x2 = p[j].x;
46                     ans_y2 = p[j].y;
47                 }
48             }
49         }
50     }
51 }
```

2번 코드-2

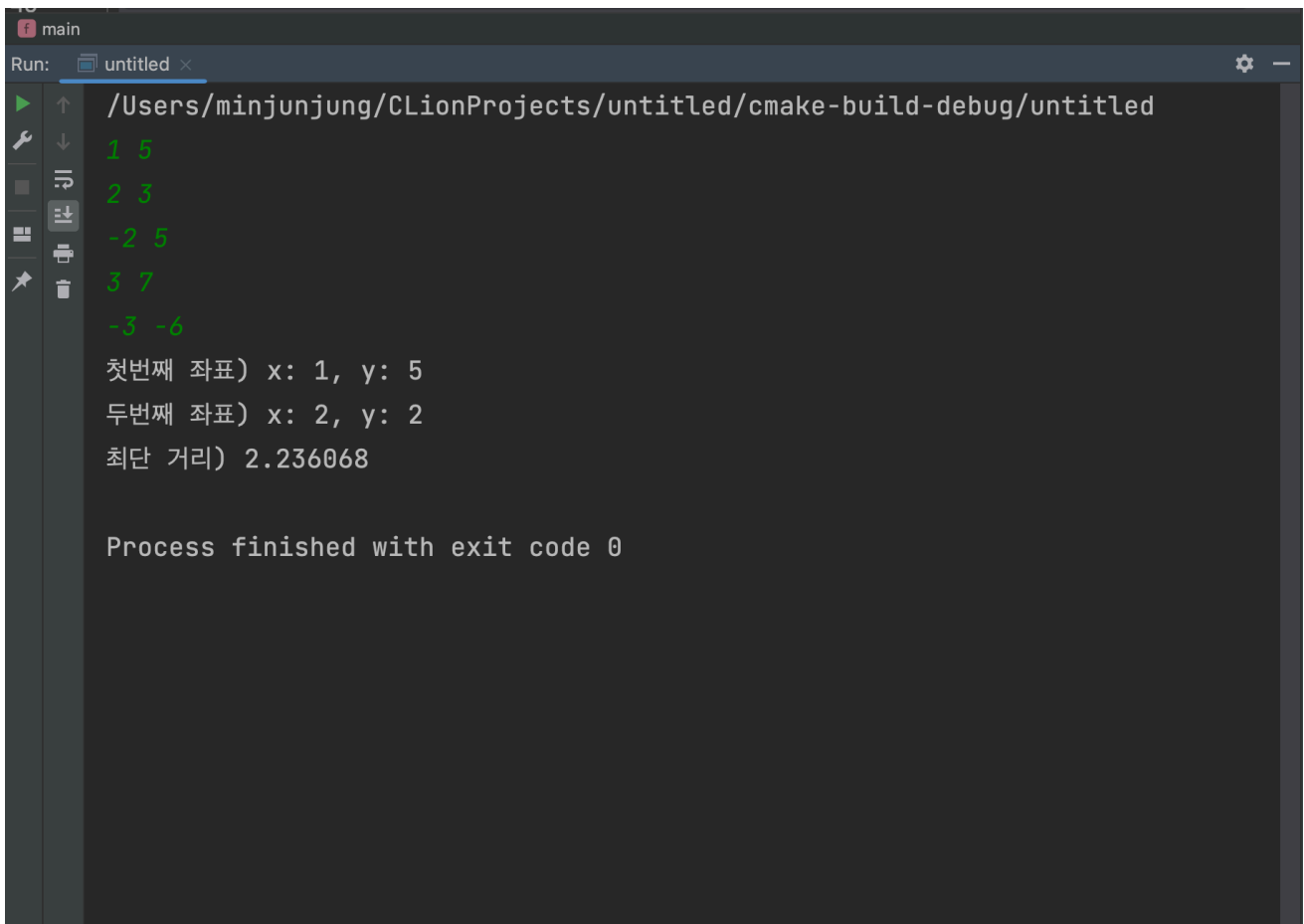
```
20
21 //2중 반복문으로 비교
22 for(int i=0; i<4; i++)
23 {
24     for(int j=i+1; j<5; j++)
25     {
26         if(i==0 && j==1)
27         {
28             //두 좌표 간의 거리가 담길 변수 선언 및 계산 첫 시행 이므로 정답 변수에 할당
29             ans = sqrt(pow(p[i].x-p[j].x,2) + pow(p[i].y-p[j].y,2));
30             ans_x1 = p[i].x;
31             ans_y1 = p[i].y;
32             ans_x2 = p[j].x;
33             ans_y2 = p[j].y;
34         }
35         else
36         {
37             double tmp; //두 좌표 간의 거리가 담길 변수 선언 및 계산
38             tmp = sqrt(pow(p[i].x-p[j].x,2) + pow(p[i].y-p[j].y,2));
39
40             if(tmp<ans) //기존 값보다 작다면 최단 거리 및 좌표값 갱신
41             {
42                 ans = tmp;
43                 ans_x1 = p[i].x;
44                 ans_y1 = p[i].y;
45                 ans_x2 = p[j].x;
46                 ans_y2 = p[j].y;
47             }
48         }
49     }
50 }
51
52 printf("첫번째 좌표) x: %d, y: %d\n",ans_x1,ans_y1);
53 printf("두번째 좌표) x: %d, y: %d\n",ans_x2,ans_y2);
54 printf("최단 거리) %f\n",ans);
55
56 return 0;
57 }
```

Structure Bookmarks

main

Run: untitled x

2번 결과 & 설명



```
Run: untitled x
/Users/minjunjung/CLionProjects/untitled/cmake-build-debug/untitled
1 5
2 3
-2 5
3 7
-3 -6
첫번째 좌표) x: 1, y: 5
두번째 좌표) x: 2, y: 2
최단 거리) 2.236068

Process finished with exit code 0
```

1. 구조체 point를 선언하고 해당 구조체는 x와 y의 좌표값을 가집니다.
2. 크기가 5인 point의 구조체 배열을 p로 선언해 줍니다.
3. 반복문과 scanf를 사용해서 구조체 배열의 각각 x, y에 입력 값을 넣습니다.
4. 2중 반복문을 통해서 각 좌표들을 사이의 거리를 모두 계산해 줍니다.
5. 두 좌표의 거리 계산 공식은 $\sqrt{\text{pow}(p[i].x - p[j].x, 2) + \text{pow}(p[i].y - p[j].y, 2)}$ 입니다.
6. 만약 초깃값 ans보다 계산된 거리의 값이 작다면 정답 좌표들과 최단거리를 갱신해 줍니다.