

C Programming

12주차 실습

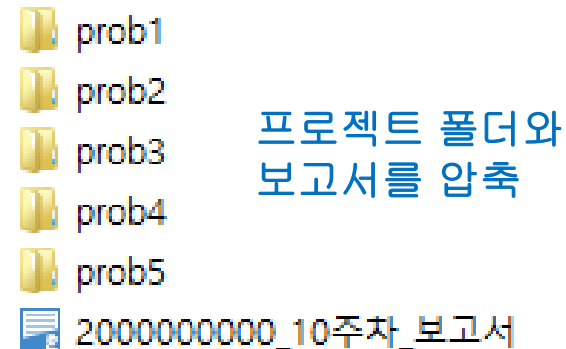
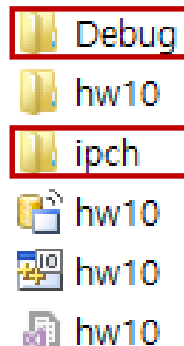


컴퓨터 공학과
이기훈 교수님



실습 운영 계획

- 실습시간에 나가는 모든 문제는 과제임
 - 프로젝트 폴더와 보고서를 압축해서 제출
 - 파일명 : C프_12주차_학번_이름.zip
 - 주석은 최대한 작성할 것
 - 문제에 있는 조건을 지키지 않을 시 감점
 - Copy 발견 시, 해당 주차 과제 0점 처리
-
- 프로젝트 폴더 제출 방법
 - 문제마다 새 프로젝트 생성
 - 프로젝트 폴더 안에 있는 ipch, Debug 폴더는 반드시 삭제





실습 운영 계획

□ 보고서 작성 요령

- 코드
 - ✓ 본인이 작성한 코드를 텍스트로 작성 또는 코드 부분만 캡처해서 첨부
- 결과화면
 - ✓ 코드 실행화면을 캡처해서 첨부
- 고찰
 - ✓ 실습 수행 중 잘 풀리지 않은 부분이나 중요한 알고리즘 등을 설명
- Reference
 - ✓ 인터넷 사이트 주소, 친구 도움 등을 문제별로 작성 (강의자료, 교재의 경우 생략 가능)

□ 제출 방법

- U-CAMPUS 로그인 -> 온라인 참여 학습 -> 과제제출
- 하드카피는 따로 받지 않음



문제 1

- 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 다섯 개의 정수는 크기가 5인 int형 배열에 저장
2. 배열에 저장된 다섯 개의 정수에 곱하기 2를 한 결과를 다시 동일한 배열에 저장
3. 단, 2번을 수행할때 반드시 포인터를 이용

결과화면

```
>> 다섯 개의 정수 입력: 1 2 3 4 5  
>> 결과(x2): 2 4 6 8 10
```

```
>> 다섯 개의 정수 입력: 2 1 0 1 2  
>> 결과(x2): 4 2 0 2 4
```



문제 2

- 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 입력받은 문장은 크기가 256인 char형 배열에 저장
2. 문자열(문장)을 입력받는 경우 `gets()` 또는 `fgets()` 함수를 이용
3. count를 위해 배열을 탐색할 때, 반드시 포인터를 이용

결과화면

```
>> 문장 입력: hello world
>> 카운트할 문자 입력: 1
>> 문자 <1> count = 3
```

```
>> 문장 입력: hello world
>> 카운트할 문자 입력: x
>> 문자 <x> count = 0
```

참고

```
void main(){
    char str[256]={0};
```

```
    gets(str); // Enter가 입력될 때까지 문자열을 입력받아 str에 저장
    printf("%s", str);
```

```
}
```



문제 3

- 다음 주어진 코드를 참고하여 사용자가 입력한 문장을 한 글자씩 차례로 출력하는 프로그램을 작성하시오.

[조건]

1. 입력받은 문장은 크기가 256인 char형 배열에 저장
2. 출력을 위해 배열을 탐색할 때, 반드시 포인터를 이용

Hint

```
Sleep(100)  
// 100ms = 0.1초간 프로그램 실행을 지연  
// # include <Windows.h> 추가
```

결과화면

```
>> 문장 입력: press any key to continue
```

```
>> 문장 입력: press any key to continue  
>> 결과: press
```

```
>> 문장 입력: press any key to continue  
>> 결과: press any k
```

```
>> 문장 입력: press any key to continue  
>> 결과: press any key to continue  
계속하려면 아무 키나 누르십시오 . . .
```



문제 4

- 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 입력받은 문장은 크기가 256인 char형 배열에 저장
2. 같은 글자로 시작하는 단어가 여러 개일 경우 가장 앞에 있는 단어 하나만 출력
3. 반복문은 while 또는 do ~ while문을 사용

결과화면

```
>> 문장 입력: Maybe you think that you can  
>> 검색할 단어의 첫 글자 입력: t  
>> think
```

```
>> 문장 입력: Maybe you think that you can  
>> 검색할 단어의 첫 글자 입력: i  
>> i로 시작하는 단어가 없습니다
```



문제 5

- ▣ 버블 소트(Bubble Sort) 프로그램을 구현하시오.

[조건]

1. swap이 일어날 때마다 list를 출력
2. 반드시 다음페이지에 주어진 코드를 이용하여 프로그램을 완성할 것
3. 주어진 코드 이외의 방법으로 구현할 경우 감점

결과화면

```
#--- Bubble Sort ---#
1. Ascending Order
2. Descending Order
3. Exit

Select: 5
Select: 1
Input the list <5>: 4 2 1 5 3
>> 2 4 1 5 3
>> 2 1 4 5 3
>> 2 1 4 3 5
>> 1 2 4 3 5
>> 1 2 3 4 5

Select: 2
Input the list <5>: 3 5 4 1 2
>> 5 3 4 1 2
>> 5 4 3 1 2
>> 5 4 3 2 1

Select: 3
계속하려면 아무 키나 누르십시오 . . .
```

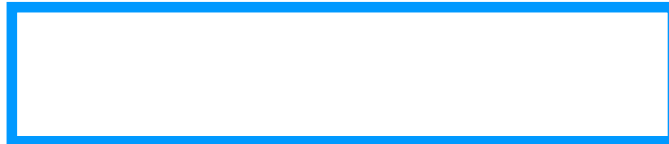



문제 5 - 참고

```
void bubble_sort(int flag, int *arr);  
void swap(int *n1, int *n2);  
int compare(int flag, int n1, int n2){  
    return flag==1? (n1>n2):(n1<n2);  
}
```

```
int main(){  
    int list[5]={0};  
    int i=0, flag=0;  
  
    printf(" #--- Bubble Sort ---#\n");  
    printf(" 1. Ascending Order\n");  
    printf(" 2. Descending Order\n");  
    printf(" 3. Exit\n\n");
```

```
do{
```



```
}while(1);
```

```
return 0;
```

```
}
```

bubble_sort() 구현
swap() 구현

swap, compare 함수는
bubble_sort 함수 안에서 사용



문제 5 - 참고

□ 버블 소트(Bubble Sort)

- 이웃한 두 원소를 비교하면서 정렬하는 정렬 알고리즘이다
- 속도는 느리지만 코드가 단순하기 때문에 자주 사용된다.

(참고) http://ko.wikipedia.org/wiki/%EA%B1%B0%ED%92%88_%EC%A0%95%EB%A0%AC

(예)

4	7	3	2
---	---	---	---

오름
차
순

4	7	3	2
4	3	7	2
4	3	2	7
4	3	2	7

더 이상
정렬 필요 X

< Step 1 >

다시 처음부터 정렬

3	4	2	7
3	2	4	7
3	2	4	7

< Step 2 >

2	3	4	7
2	3	4	7

< Step 3 >