

## 기초프로그래밍 과제#4

### ◆ 과제 #4 범위

- 제 10장, 11장 배열

### ◆ 과제공지: 4월 30일(수)

### ◆ 제출방법: 담당 교수님께서 공지하신대로

### ◆ 제출마감: 5월 15(목) 11:59

### ◆ 총점 30점 (문제별 점수 표기함)

### ◆ 주의 사항:

- turnin으로 제출 후 hw.pdf 파일에 작성한 소스 코드가 출력되었는지 확인하세요.
- 제출 마감 시간 이후, 추가 제출은 받지 않습니다.
- 부정행위(인터넷 자료 이용 및 타인 숙제 복사 등의 제출)는 0점 처리됩니다.
- 제출된 코드가 컴파일 혹은 프로그램 실행 불가 시, 0점 처리됩니다.

1. (10점) 사용자로부터 두 개의 문자열과 메뉴 숫자 한 개를 입력 받고 아래의 조건을 실행하는 프로그램을 작성하세요. 단, 다음의 조건을 모두 만족해야 합니다.

▪ 입력 조건

- 프로그램 시작 시점에 문자열 두 개, 숫자 한 개를 입력 받음
- 문자열 두 개 이하 또는 이상이 입력된 경우, "Wrong\_start" 메시지 출력 후 프로그램 종료
- 문자열 위치에 숫자 입력 또는 숫자 위치에 문자열이 입력되는 경우, "Wrong\_arguments" 출력 후 프로그램 종료
- 두 문자열 입력과 메뉴 숫자가 1이면, 두 개의 문자열을 비교하고 같은 경우, "Same" 출력, 다른 경우, "Different" 출력 후 프로그램 종료
- 두 문자열 입력과 메뉴 숫자가 2이면, 두 번째 문자열을 기준으로 두 문자열 중 첫번째로 다른 문자를 출력하고 프로그램 종료
- 두 문자열 입력과 메뉴 숫자가 3이면, 두 번째 문자열 기준으로 두 문자열의 다른 모든 단어를 출력하고 프로그램 종료.

▪ 함수 조건

- 메뉴 숫자가 1인 경우 실행될 dgu\_strcmp() 함수를 구현하세요.
- 메뉴 숫자가 2인 경우, 실행될 dgu\_find() 함수를 구현하세요.
- 메뉴 숫자가 3인 경우, 실행될 dgu\_diff() 함수를 구현하세요.

▪ 힌트

- 프로그램 시작 시점의 프로그램에 데이터 전달은 main (int argc, char \*argv[]) 사용
- 함수 전달 인자는 char str1[] 사용
- \* 예: void dgu\_strcmp(char str1[], char str2[])

▪ 제한 조건

- 컴파일된 실행 파일명을 "dgustr"로 지정하세요. (gcc -o 옵션 사용)
- string 관련 라이브러리 사용 불가 (예: strcmp, strcat, strcpy, 등)

▪ 채점 기준

- 실행 결과가 주어진 조건을 모두 만족하고 정확하게 동작하는가? [10점]
- dgu\_strcmp, dgu\_find, dgu\_diff을 모두 구현하지 못했는가? [-5점]

**실행 예시 1: 사용자 입력은 파란색으로 표기함**

```
linuxserver1:~$./dgustr a a
Wrong_start
linuxserver1:~$./dgustr a a a a
Wrong_start
linuxserver1:~$./dgustr a a a
Wrong_arguments
linuxserver1:~$./dgustr a 1 1
Wrong_arguments
```

**실행 예시 2: 사용자 입력은 파란색으로 표기함**

```
linuxserver1:~$./dgustr ab abc 1
Different
linuxserver1:~$./dgustr abc abc 1
Same
```

**실행 예시 2: 사용자 입력은 파란색으로 표기함**

```
linuxserver1:~$./dgustr ab abc 2
c
linuxserver1:~$./dgustr abcdef abcdef 2
f
linuxserver1:~$./dgustr 1234567 12345678 2
8
linuxserver1:~$./dgustr abcdef abcdef 2
linuxserver1:~$ // 정상 종료
```

**실행 예시 3: 사용자 입력은 파란색으로 표기함**

```
linuxserver1:~$./dgustr abcd abcee 3
ee
linuxserver1:~$./dgustr 123 1a3b5c 3
ab5c
linuxserver1:~$./dgustr a a1b3c5 3
1b3c5
linuxserver1:~$./dgustr abcfghi ab 3
linuxserver1:~$ // 정상 종료
linuxserver1:~$./dgustr 123456 1a3b5c 3
abc
linuxserver1:~$./dgustr abcfghi a1c 3
1
linuxserver1:~$./dgustr a1c a1c 3
linuxserver1:~$ // 정상 종료
```

2. (20점) 사용자로부터 입력 받은 정보를 기반으로 각 자동차가 교량을 통과한 시간을 출력하세요. 단, 아래의 자동차와 교량은 아래의 조건에 만족해야함.

▪ 조건

- 교량은 단방향 두 개의 차로 (즉, 편도 2차로)를 가지고 있음
- 교량에는 최대 두 대의 차량만 동시에 진입이 가능하고, 교량에 위치한 모든 차량의 무게가 교량이 수용 가능한 무게 (고정값: 10)보다 큰 경우, 다음 차량이 진입할 수 없음.
- 자동차는 Time이 1 증가할 때만 진입 여부를 판단할 수 있으며, 두 차로 모두 진입 가능할 경우 첫 번째 차로의 차량이 우선적으로 진입 가능함.

▪ 입력 조건

- 첫번째 차로를 통과할 차량의 정보를 숫자로 입력 받음 (최소 5, 최대 100)
- 통과할 차량의 정보의 범위에 오류가 있는 경우, 사용자로부터 다시 입력 받음
- 첫번째 차로의 각 차량의 이름을 입력 받음
- 첫번째 차로의 각 차량의 무게를 입력 받음 (최소 1, 최대 10)
- 두번째 차로를 통과할 차량의 정보를 숫자로 입력 받음 (최소 5, 최대 100)
- 통과할 차량의 정보의 범위에 오류가 있는 경우, 사용자로부터 다시 입력 받음
- 두번째 차로의 각 차량의 이름을 입력 받음
- 두번째 차로의 각 차량의 무게를 입력 받음 (최소 1, 최대 10)

▪ 출력 조건

- Time이 증가할 때마다 현재 교량의 상태를 출력
- 교량을 나오는 차량의 이름과 무게 출력
- 모든 차량이 통과한 후 통과한 전체 차량의 무게와 시간을 출력

▪ 제한 조건

- 교각을 통과할 차량의 입력 범위의 오류가 있는 경우, 재 입력 받음
- 차량 무게의 범위의 오류가 있는 경우, 재 입력 받음
- 배열로 구현
- 차량의 길이는 영문자 대문자 또는 소문자 1자로 제한함

▪ 채점 기준

- 실행 결과가 주어진 조건을 모두 만족하고 정확하게 동작하는가? [20점]
- 배열로 구현하지 못한 경우 [-10점]
- 출력 형식이 예시와 다른 경우 [-5점]

**실행 예시 1: 사용자 입력은 파란색으로 표기함**

```
1st Max Number: 500
Wrong number
1st Max Number: 1
Wrong number
1st Max Number: 5
1st Name: A B C D E
1st Weight: 5 4 7 3 11
Wrong weight
1st Weight: 5 4 7 3 8
```

**실행 예시 2: 사용자 입력은 파란색으로 표기함**

```
1st Max Number: 5
1st Name: A B C D E
1st Weight: 5 4 7 3 8
2nd Max Number: 5
2nd Name: a b c d e
2nd Weight: 2 1 7 1 2
(time 0) 1st [ - - ]
(time 0) 2nd [ - - ]
(time 1) 1st [ A - ]
(time 1) 2nd [ a - ]
(time 2) 1st [ - A ]
(time 2) 2nd [ b a ]
(time 3) 1st [ B - ] out: A (5)
(time 3) 2nd [ - b ] out: a (2)
(time 4) 1st [ - B ]
(time 4) 2nd [ - - ] out: b (1)
(time 5) 1st [ C - ] out: B (4)
(time 5) 2nd [ - - ]
(time 6) 1st [ D C ]
(time 6) 2nd [ - - ]
(time 7) 1st [ - D ] out: C (7)
(time 7) 2nd [ c - ]
(time 8) 1st [ - - ] out: D (3)
(time 8) 2nd [ d c ]
(time 9) 1st [ E - ]
(time 9) 2nd [ - d ] out: c (7)
(time 10) 1st [ - E ]
(time 10) 2nd [ e - ] out: d (1)
(time 11) 1st [ - - ] out: E (8)
(time 11) 2nd [ - e ]
(time 12) 1st [ - - ]
(time 12) 2nd [ - - ] out: e (2)
Total weight: 40
Elapsed time: 13
```