

기술인증테스트 시스템&솔루션개발 실기형 문제지

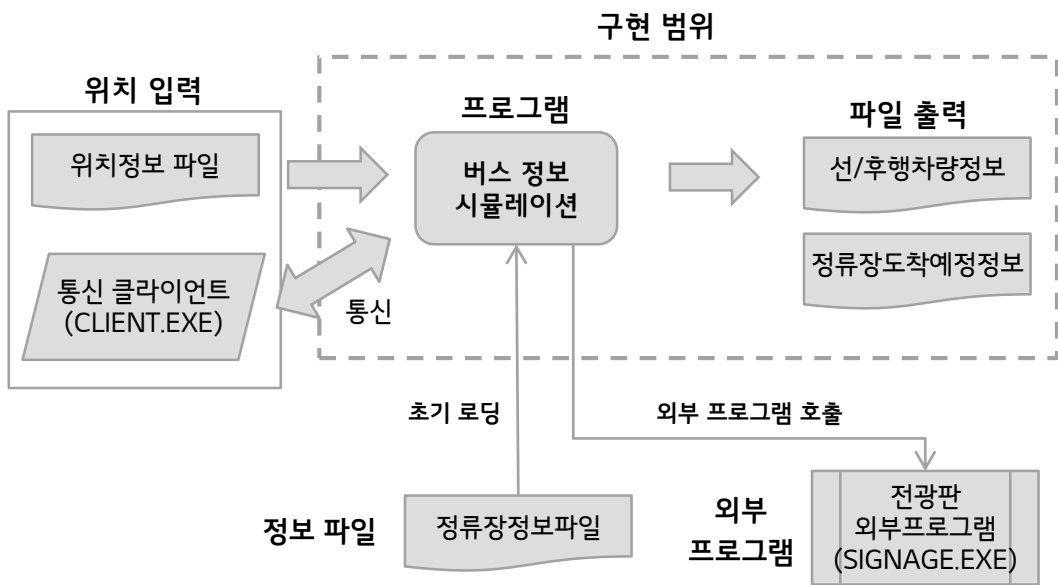
- 문항설명 -

[2019년 #차]

사번	성명
유의 사항	<p>1. 공정한 평가를 위해 동료들 도와주는 행위, 보여주는 행위를 금지하고 있습니다.</p> <p>2. 부정행위 적발 시, 응시한 평가는 0점 처리됩니다.</p> <p>3. 본 시험지는 응시장 외부로 유출할 수 없으며, 시험 종료 후 감독관에게 제출해야 합니다.</p>

1. 버스 정보 시뮬레이션

개요	<p>버스 정보 시뮬레이션 구현을 통해</p> <p>요구사항 분석, 파일처리, 데이터관리, 외부 프로그램 호출, 통신처리, 병렬 처리 등의 기술역량 및 프로그램 구현 역량을 측정하기 위한 문제입니다.</p>
설명	<p>본 프로그램은 파일 또는 통신 클라이언트로부터 버스의 위치 정보를 입력 받아, 선/후행차량정보와 정류장도착예정정보, 승객도착예정시각 등을 시뮬레이션하여 파일과 외부프로그램으로 출력하는 프로그램입니다.</p>

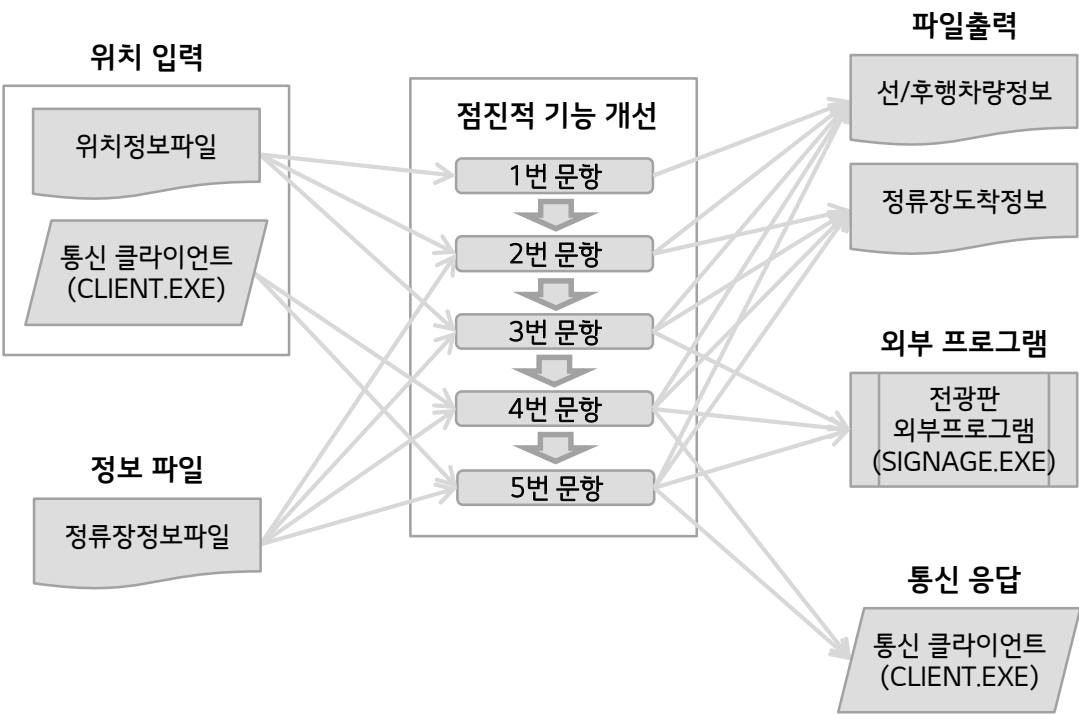


- [기능 요약]
- 위치정보를 파일 또는 통신 클라이언트로부터 입력 받음
 - 입력된 위치정보와 정류장정보를 이용하여 선/후행차량정보와 정류장도착예정정보 계산
 - 위치정보가 손실되는 경우 차량의 최종 속도와 정류장 구간 제한속도를 고려하여 최종 위치 계산
 - 차량의 최종 속도와 정류장 구간 제한속도를 고려하여 정류장도착예정시각 계산 및 외부프로그램 출력
 - 차량의 최종 속도와 정류장 구간 제한속도를 고려하여 승객의 정류장도착예정시각 계산 요청에 대한 응답 처리
 - 차량의 최종 속도와 정류장 구간 제한속도를 고려하여 선/후행차량시간 계산 요청에 대한 응답 처리

주의 사항

- 실행 결과로 평가하고 부분점수는 없으므로 아래사항을 필히 주의해야 함
- 구현된 프로그램은 실행 완결성 필수 (명확한 실행&종료 처리, 정확한 결과 출력, 통상의 실행 시간)
 - 소 문항별 결과 검수 필수 (선행문항 오류 시, 후속문항 전체에 오류가 발생할 수 있음)
 - 제시된 조건이 없는 한 선행요구사항 유지 필수 (소 문항별 입출력 관계도 참고)
 - 프로그램 실행 위치 및 실행결과출력 (위치, 파일명, 데이터포맷)은 요구사항과 정확히 일치 필수
 - 제시된 모든 위치는 상대경로 사용 필수 (프로그램 실행 위치 기준)
 - 종료조건에 맞는 자동종료 처리 필수 (불필요한 종료방해처리 (pause/입력대기/등) 하면 안됨)
 - 모든 문자는 대문자 처리 필수

문항 관계



문제

아래 제시된 문항은 문항번호가 증가할 수록 점진적 개선을 요구하는 방식으로 구성되어 있으며, 제시된 문항번호 별로 각각 **구현된 소스와 컴파일 된 실행파일을 제출**하시오.
cf) 1번 구현 → 1번 소스복사 → 2번 구현 → 2번 소스복사 → ...
<주의> 형식(별도 형식정보 문제지)을 준수하여 평가 대상을 정확하게 구현할 것

1. 위치정보파일을 읽어서 차량별 최근접 선행/후행 차량의 거리를 계산하여 파일에 출력하시오. (20점)
- 선/후행차량 거리 계산 : 위치정보파일(LOCATION.TXT, 2page 형식1)을 읽어 최종 "PRINT" 커맨드에서 차량별 최근접 선행/후행 차량을 찾고 거리 계산

- 파일출력 처리 : 계산된 선/후행차랑거리를 차량번호 순서대로 파일(PREPOST.TXT)에 출력 (4page 형식3)
- 평가대상 : 프로그램 정상 실행, 출력된 파일 내용, 프로그램 자동 종료

(위치정보 예시) 위치정보파일(LOCATION.TXT) - 2page 형식1

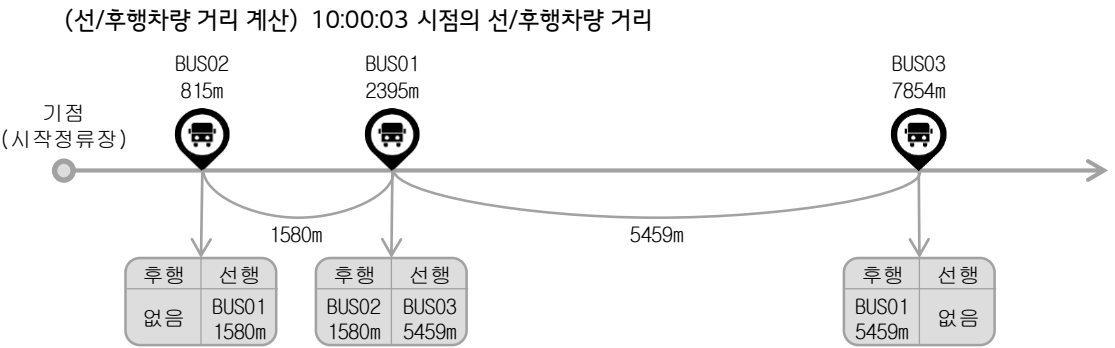
10:00:00#BUS01,02345#BUS02,00800#BUS03,07843
10:00:01#BUS01,02362#BUS02,00805#BUS03,07847
10:00:02#BUS01,02378#BUS02,00810#BUS03,07850
10:00:03#BUS01,02395#BUS02,00815#BUS03,07854
PRINT

← "PRINT" 커맨드 (10:00:03 시점)

(출력 예시) 선/후행차랑정보파일(PREPOST.TXT) - 4page 형식3

10:00:03#BUS01#BUS03,05459#BUS02,01580
10:00:03#BUS02#BUS01,01580#N0BUS,00000
10:00:03#BUS03#N0BUS,00000#BUS01,05459

} 10:00:03 시점의 선/후행차랑거리



문제

2. 위 1번 문항까지 구현된 내용을 기준으로, 아래사항을 추가로 반영하여 구현하시오. (15점)

- 위치정보형식 변경 : 위치정보파일(LOCATION.TXT)에서 차량의 대수가 3대에서 N대(1대~99대)로 변경 (2page 형식1)
 - 정류장도착예정 거리 계산 추가 : 정류장정보파일(STATION.TXT, 3page 형식2) 로딩 후 위치정보파일을 읽어 최종 "PRINT" 커맨드에서 정류장별 최근접 도착예정차량을 찾고 거리 계산
 - 파일출력 추가 : 정류장도착예정 거리 계산 결과를 정류장번호 순서대로 파일(ARRIVAL.TXT)에 출력 (4page 형식4)
- 평가대상 : 프로그램 정상 실행, 출력된 파일 내용, 프로그램 자동 종료

(위치정보 예시) 위치정보파일(LOCATION.TXT) - 2page 형식1

```
10:00:00#BUS01,02345#BUS02,00800#BUS03,07843#BUS04,08504
10:00:01#BUS01,02362#BUS02,00805#BUS03,07847#BUS04,08508
10:00:02#BUS01,02378#BUS02,00810#BUS03,07850#BUS04,08511
10:00:03#BUS01,02395#BUS02,00815#BUS03,07854#BUS04,08515
PRINT
```

← "PRINT" 커맨드 (10:00:03 시점)

(정류장정보 예시) 정류장정보파일(STATION.TXT) - 3page 형식2

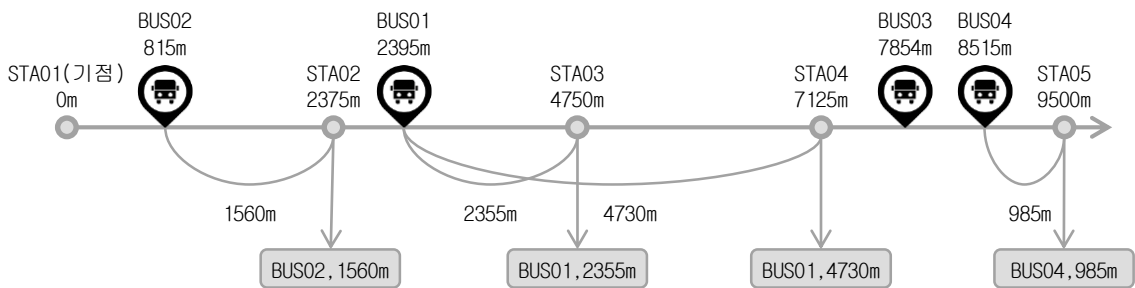
```
STA01#00000
STA02#02375
STA03#04750
...
```

(출력 예시) 정류장도착예정정보파일(ARRIVAL.TXT) - 4page 형식4

```
10:00:03#STA01#NOBUS,00000
10:00:03#STA02#BUS02,01560
10:00:03#STA03#BUS01,02355
...
```

} 10:00:03 시점의 정류장도착예정정보

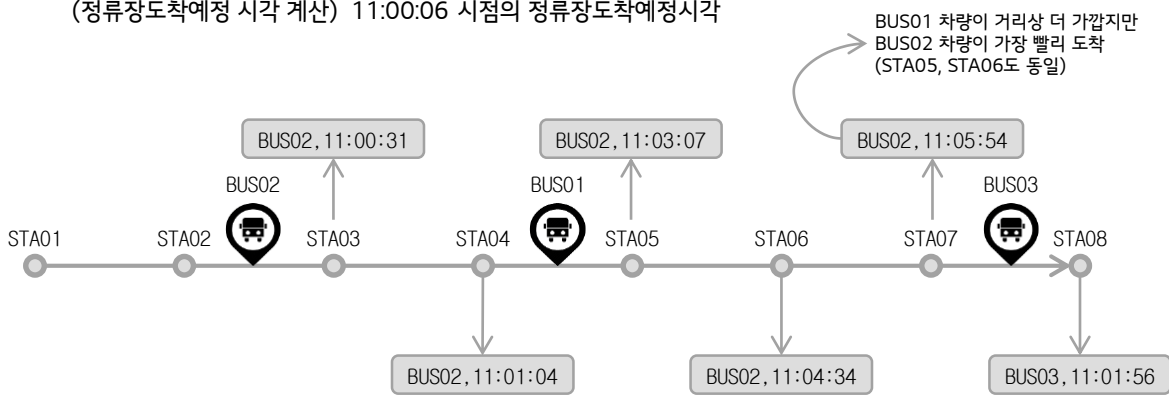
(정류장도착예정 거리 계산) 10:00:03 시점의 정류장도착예정정보



문제

3. 위 2번 문항까지 구현된 내용을 기준으로, 아래사항을 추가로 반영하여 구현하시오. (15점)
- 위치정보형식 변경 : 위치정보파일(LOCATION.TXT)에서 위치정보 손실 발생 (2page 형식1)
 - 정류장정보파일 변경 : 정류장정보파일(STATION.TXT)에 정류장 구간 제한속도(km/h) 추가 (3page 형식2)
 - 위치정보 손실을 고려한 최종 위치 계산 : 위치정보 손실이 발생하는 경우 최종 위치정보를 이용하여 차량의 최종속도를 계산하고, 정류장 구간 제한속도와 차량 최종속도를 이용하여 최종 위치 계산 (5page 형식5)
 - 정류장도착예정시간 계산 추가 : 정류장 구간 제한속도와 차량 기본속도를 이용하여 최종 "PRINT" 커맨드에서 정류장별로 가장 빨리 도착하는 차량을 찾고 정류장도착예정시간 계산 (6page 형식6)
 - 출력정보 추가 : 외부 프로그램(SIGNAGE.EXE)에 정류장도착예정시간 출력 (7page 형식7)
- 평가대상 : 프로그램 정상 실행, 출력된 파일 내용, 프로그램 자동 종료

(정류장도착예정 시각 계산) 11:00:06 시점의 정류장도착예정시간



문제

4. 위 3번 문항까지 구현된 내용을 기준으로, 아래사항을 추가로 반영하여 구현하시오. (20점)
- 위치정보입력 방식 변경 : 통신 클라이언트(CLIENT.EXE)로부터 위치정보를 수신하여 입력하는 방식
(8page 형식8, 9page 형식9)

- 모바일 "PRINT" 커맨드 처리 : 모바일 접속으로부터 "PRINT" 커맨드 수신하여 출력 처리
(8page 형식8, 9page 형식9)

1) 선/후행차량정보파일 기록

2) 정류장도착예정정보파일 기록

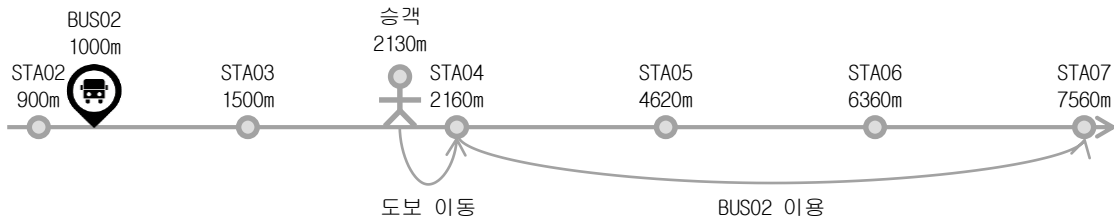
3) 정류장도착예정시각 계산 및 외부프로그램 호출

3번 문항에서 이미 구현한 기능

- 모바일 승객도착예정시각 요청/응답 추가 : 모바일 접속으로부터 승객도착예정시각 요청 수신시
승객도착예정시각 계산하여 응답 (11page 형식10)

→ 평가대상 : 프로그램 정상 실행, 출력된 파일 내용, 프로그램 종료 없음

(승객도착예정시각 계산) 11:00:06 시점의 승객도착예정시각

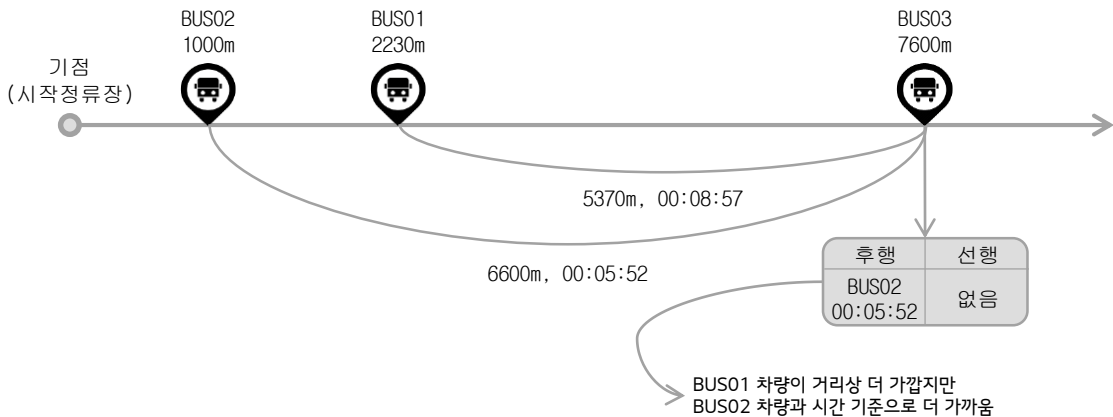


	출발시각	STA03 도착예정시각	STA04 도착예정시각	STA05 도착예정시각	STA06 도착예정시각	STA07 도착예정시각
BUS02	11:00:06	11:00:31	11:01:04	11:03:07	11:04:34	11:05:54
승객	11:00:06	N/A	11:00:36 (도보) 11:01:04 (BUS02 승차)	11:03:07 (BUS02)	11:04:34 (BUS02)	11:05:54 (BUS02)

- 문제
5. 위 4번 문항까지 구현된 내용을 기준으로, 아래사항을 추가로 반영하여 구현하시오. (10점)
- 차량 선/후행차량시간 요청/응답 추가 : 차량 접속으로부터 선/후행차량시간 요청 수신시 요청 차량과 시간 기준으로 가장 가까운 선/후행차량을 찾고 시간을 계산하여 응답 수행 (9page 형식9, 12page 형식11)

→ 평가대상 : 프로그램 정상 실행, 출력된 파일 내용, 프로그램 종료 없음

(선/후행차량 시간 계산) 11:00:06 시점의 BUS03 차량의 선/후행차량시간



폴더 정보

- ※ 프로그램 및 파일 위치 정보 (실행위치 기반 상대경로 사용 필수)
- 구현할 프로그램 위치 및 실행 위치 : 각 문항 홈 (SUB1 / SUB2 / SUB3 / SUB4 / SUB5)
 - 입력 파일 위치 : 각 문항 홈 아래 **INFILE** 폴더 (LOCATION.TXT, STATION.TXT)
 - 출력 파일 위치 : 각 문항 홈 아래 **OUTFILE** 폴더 (PREPOST.TXT, ARRIVAL.TXT)
 - 제공되는 통신 클라이언트 프로그램 위치 및 파일명 : 각 문항 홈 아래 CLIENT.EXE
 - 제공되는 정류장도착예정시각출력 프로그램 위치 및 파일명 : 각 문항 홈 아래 SINAGE.EXE
 - 자가 검수용 참고 파일명 : OUTFILE폴더 내 CMP_*.TXT
- * 제공되는 파일들은 문항에 따라 다를 수 있음

실행 방식

- ※ 구현할 프로그램 형식
- 프로그램 형태 : 콘솔(Console) 프로그램
 - 프로그램 파일명 : SP_TEST
 - 실행 방식(문항1~3) : 콘솔 실행→결과처리→자동종료
- C:\W>SP_TEST<엔터키> ← 콘솔 실행 / 결과처리 후 프로그램 자동 종료
C:\W>
- 실행방식(문항4~5) : 콘솔 실행→통신연결대기→통신처리 (종료 없음)
- C:\W>SP_TEST<엔터키>

제공 정보 및
제출 정보

- ✓ 각 언어별 제공파일 압축 해제 후 자동 생성된 폴더 사용 필수
 - ✓ 제공되는 주요 내용
 - 입력 및 출력 샘플 파일
 - 전광판 실행파일(SIGNAGE.EXE) / 통신 클라이언트 프로그램 실행파일 (CLIENT.EXE)
 - 제출시 사용할 문항별 폴더 구조
 - ✓ 제출 파일 및 폴더 상세 내용 (각 언어별 실기 가이드 참고)
- <주의사항>
- 제출 파일 관련 내용 (폴더위치, 파일명, 프로그램명 등) 이 틀린 경우 및 상대경로를 사용하지 않은 경우에는 평가 시 불이익이 발생할 수 있으므로 반드시 요구되는 내용과 일치시켜 제출해야 함.

테스트 방법

※ 자가 검수를 위해 제공되는 샘플은 검수용 데이터와 다를 수 있음

검수를 위한 샘플 결과 파일은 각 문항 출력 폴더(OUTFILE)에 사전 제공됨

- [문항1]
- SP_TEST를 실행한 후 출력 결과 파일(PREPOST.TXT)을 샘플 결과 파일(CMP_PREPOST.TXT)와 동일한지 비교
- [문항2]
- SP_TEST를 실행한 후 출력 결과 파일들(PREPOST.TXT, ARRIVAL.TXT)을 샘플 결과 파일들(CMP_PREPOST.TXT, CMP_ARRIVAL.TXT)과 동일한지 비교.
- [문항3]
- SP_TEST를 실행한 후 출력 결과 파일들(PREPOST.TXT, ARRIVAL.TXT, SIGNAGE.TXT)을 샘플 결과 파일들(CMP_PREPOST.TXT, CMP_ARRIVAL.TXT, CMP_SIGNAGE.TXT)과 동일한지 비교.
- [문항4]
- SP_TEST를 실행한 후 CLIENT.EXE 실행.
출력 결과 파일들(PREPOST.TXT, ARRIVAL.TXT, SIGNAGE.TXT, MOBILE.TXT)을 샘플 결과 파일들(CMP_PREPOST.TXT, CMP_ARRIVAL.TXT, CMP_SIGNAGE.TXT, CMP_MOBILE.TXT)과 동일한지 비교.
- [문항5]
- SP_TEST를 실행한 후 CLIENT.EXE 실행.
출력 결과 파일들(PREPOST.TXT, ARRIVAL.TXT, SIGNAGE.TXT, MOBILE.TXT, BUS03.TXT)을 샘플 결과 파일들(CMP_PREPOST.TXT, CMP_ARRIVAL.TXT, CMP_SIGNAGE.TXT, CMP_MOBILE.TXT, CMP_BUS03.TXT)과 동일한지 비교.

