

기술인증테스트 Python개발-B 실기형 문제지

[2017년 @차]

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



실기테스트 구현 시 주의사항

- ✓ 별도 제공된 실기가이드에 따라 실습 환경을 구성합니다.
- ✓ 실습 환경을 구성하고, 제공된 파일을 Python IDLE에서 작동하는 것도 시험의 일부입니다.
실기가이드를 따라 하시면 됩니다. 환경 구성에 대해 감독관은 질문을 받지 않습니다.
- ✓ 본 실기테스트는 Python 3.X 버전을 기준으로 작성합니다.
- ✓ 문항에 대한 질문은 받지 않습니다. 단, 오타자 및 문항오류에 대해서는 문의하실 수 있습니다.
- ✓ 제공된 파일명과 함수의 이름 및 형태를 변경할 수 없습니다.
 - ➔ 채점 툴(tool)을 통한 자동 채점이 이루어집니다.
 - 제공된 파일의 원형을 변경하시면 채점이 되지 않습니다.
- ✓ 구현 내용은 반드시 주어진 함수 내에서 작성해야 합니다.

1. 자동차 통행료 계산하기

개요 여러 대의 자동차가 지나갈 때 전체 통행료 및 면제 차량 대수를 계산하는 프로그램을 작성한다.

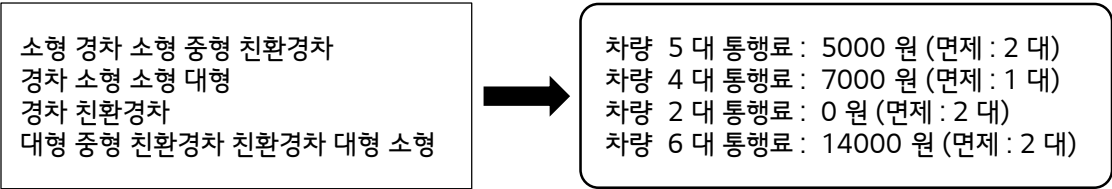
문항설명 차량이 톨게이트를 통행할 때 아래 표와 같이 통행료를 지불해야 한다.
차량의 종류는 대형/중형/소형/경차/친환경차 등 5종류만 존재한다고 가정한다.

차량종류	대형	중형	소형	경차	친환경차
통행료(원)	5000	3000	1000	면제	면제

아래와 같이 입력파일의 각 행을 읽어서, 각 행 별로 전체 통행료 및 면제차량 대수를 화면에 출력한다.

입력 파일

결과 화면



- 1. 제공된 입력 파일(data03_01.txt)은 각 줄에 통과하는 자동차의 목록이 쓰여 있다.
- 2. 입력 파일의 각 줄을 읽어서 리스트에 저장한다. (제공)
- 3. 표에 주어진 차량 종류 별 통행료를 확인하여 라인 별로 전체 통행료와 면제 차량 대수를 계산한다.

- 요구사항**
- 1. 입력 값이 잘못 입력되는 경우는 고려하지 않는다.
즉, 입력 파일에는 대형, 중형, 소형, 경차, 친환경차 만 있다.
 - 2. 경차와 친환경차는 통행료가 없지만, 면제 차량 대수를 카운팅해야 한다.

- 제공 파일**
- ✓ 제공된 소스폴더(LCP_PythonTest03WTest03_01)를 Open하여 알맞은 함수에 내용 구현
 - ✓ 코딩하도록 지정한 부분 외에는 절대 수정하면 안됨
 - ✓ 소스파일 내에서 data03_01.txt 파일을 읽어서 사용하도록 코딩되어 있음. (변경하지 말 것)
- ※ 소스파일 Open 방법은 "실기 가이드" 참고

- 배점
- 아래 제시된 기능의 구현 완성도에 따라 점수를 부여함
(단, 컴파일 에러 발생 시 점수가 부여되지 않으며, 제시된 요구사항과 다를 경우 감점 발생)
1. 정확한 결과의 출력과 합리적인 문제 해결 처리 방식(8점)

2. 리스트 접근 방식(2점)

3. 조건문, 반복문의 정확한 사용(2점)

실행 예시 1

---- Line 1 ----
차량 5 대 통행료 : 5000 원 (면제 : 2 대)

---- Line 2 ----
차량 4 대 통행료 : 7000 원 (면제 : 1 대)

---- Line 3 ----
차량 2 대 통행료 : 0 원 (면제 : 2 대)

---- Line 4 ----
차량 6 대 통행료 : 14000 원 (면제 : 2 대)

←

※ 입력 파일 내용

소형 경차 소형 중형 친환경차
경차 소형 소형 대형
경차 친환경차
대형 중형 친환경차 친환경차 대형 소형

실행 예시 2

---- Line 1 ----
차량 6 대 통행료 : 0 원 (면제 : 6 대)

---- Line 2 ----
차량 7 대 통행료 : 15000 원 (면제 : 0 대)

←

※ 입력 파일 내용

친환경차 친환경차 경차 경차 경차 경차
소형 소형 소형 대형 중형 소형 중형

2. L백화점 우수고객 선발하기

개요

L 백화점에서는 고객의 구매 내역을 관리하고 있다. 총 구매금액이 높은 고객 3명을 선발하는 프로그램을 작성한다.

문항설명

고객의 구매 내역은 별도의 순서 없이 저장되어 있다.
각 고객이 구매한 내역의 구매금액을 모두 합하여 오른쪽 표와 같이 구매금액이 큰 순서대로 등수를 매긴다.

[구매 내역]

고객 아이디	구매금액(천원)
AAA	50
BBB	330
CCC	20
AAA	70
AAA	80
BBB	20
DDD	5
EEE	200

[우수고객 선발 결과]

고객 아이디	총 구매액	등수
BBB	350	1
AAA	200	2
EEE	200	2
CCC	20	4
DDD	5	5

- 요구사항
1. 구매 내역은 파일 형태로 제공된다. 파일에는 '고객 아이디', '구매 금액' 순서대로 저장되어 있다.

2. 우수 고객 선발 결과는 2차원 리스트 형태로 반환한다.
예) [['BBB', 350, 1], ['AAA', 200, 2], ['EEE', 200, 2] ~~~]

3. 우수 고객의 등수를 매길 때, 총 구매액이 동일한 경우에는 동일한 등수를 부여한다. 이 때, 고객 아이디 순서는 상관 없다. 동일한 등수의 인원이 여러 명인 경우, 다음 등수는 인원 수만큼 건너 뛴 후 부여한다.
예) 위 문항 설명의 예시와 같이 2등이 2명인 경우, 다음 등수는 3등이 아니라 4등이다.

- 제공 파일
- ✓ 제공된 소스폴더(LCP_PythonTest03WTest03_02)를 Open하여 알맞은 함수에 내용 구현

✓ 코딩하도록 지정한 부분 외에는 절대 수정하면 안됨

✓ 소스파일 내에서 data03_02.txt 파일을 읽어서 사용하도록 코딩되어 있음. (변경하지 말 것)

※ 소스파일 Open 방법은 “실기 가이드” 참고

배점

아래 제시된 기능의 구현 완성도에 따라 점수를 부여함
(단, 컴파일 에러 발생 시 점수가 부여되지 않으며, 제시된 요구사항과 다를 경우 감점 발생)

1. 정확한 결과의 출력과 합리적인 문제 해결 처리 방식(10점)
2. 리스트 조작 기능(3점)
3. 데이터 정렬 및 동일 등수 처리 기능(3점)

실행 예시

**** [우수 고객 선발 결과] ****

----- 1 ॥-----

고객ID	총구매액	고객 순위
BBB	350	1
AAA	200	2
EEE	200	2
CCC	20	4
DDD	5	5

----- 2 회 -----

고객ID	총구매액	고객 순위
FFF	1500	1
BBB	350	2
EEE	200	3
AAA	200	3
GGG	200	3
CCC	20	6
DDD	5	7

----- 3 회 -----

고객ID	총구매액	고객 순위
CCC	789	1
EEE	260	2
RRR	200	3
GGG	200	3
YYY	175	5
BBB	107	6
DDD	35	7

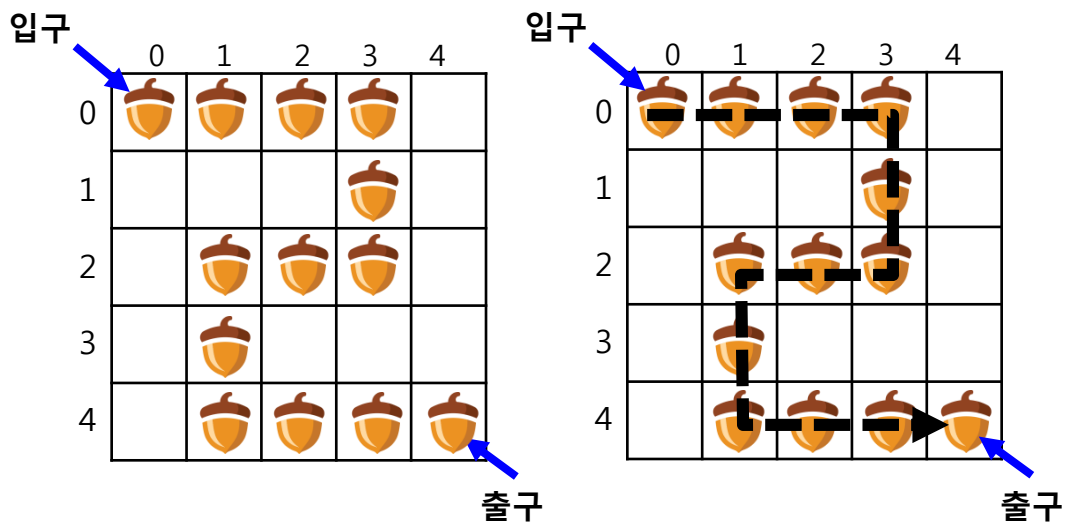
3. 다람쥐의 도토리 줍기

개요

다람쥐가 왼쪽 위 입구에서 오른쪽 아래 출구까지 찾아가는 길을 찾는다. 다람쥐는 도토리를 주우면서 출구까지 이동 한다.

문항설명

아래 그림과 같은 지도에서 다람쥐가 왼쪽 위 입구(0,0)에서 오른쪽 아래 출구(4,4)로의 탈출 경로를 찾는 프로그램을 작성한다. 그림에는 도토리가 표시되어 있고, 빈 칸은 도토리가 없다.
다람쥐는 도토리를 주으며 상하좌우로 한번에 한 칸씩 이동한다. (빈 칸으로는 이동할 수 없음)



요구사항

1. 위의 예에서 다람쥐가 이동할 수 있는 경로를 점선으로 표시했으며, 이동거리는 입구와 출구를 포함해서 13칸을 이동했다. 이동한 칸 수, 경로, 모은 도토리의 개수를 출력하는 프로그램을 작성해야 한다.

2. 위의 왼쪽 그림을 리스트로 표현하면 아래와 같다. 아래는 다람쥐가 출구를 찾아가기 전의 방의 상태(리스트)다. 0은 도토리가 없는 곳이므로 다람쥐가 지나갈 수 없다. 도토리가 있는 곳은 숫자로 표시했으며, 숫자는 도토리의 개수이다. 이 숫자 부분이 다람쥐가 지나갈 수 있는 길이다.
[1, 1, 2, 3, 0]
[0, 0, 0, 1, 0]
[0, 5, 1, 3, 0]
[0, 1, 0, 0, 0]
[0, 2, 3, 3, 1]

위 예에서 다람쥐는 (0, 0) -->(0, 1) -->(0, 2) -->(0, 3) -->(1, 3) -->(2, 3) -->(2, 2) -->(2, 1) -->(3, 1) -->(4, 1) -->(4, 2) -->(4, 3) -->(4, 4) 순서로 13칸을 이동한다. 이동하면서 주운 도토리의 개수는 27이 된다.

3. 제공되는 방 구조는 항상 5x5 이상의 정방형으로 구성되어 있다. 데이터 파일을 읽어서 ROOM 리스트에 저장되도록 제공된다. 또한 입구에서 출구까지 가는 길(=도토리가 연결되어서 떨어져 있는 길)은 반드시 1개만 존재하며, 반드시 입구에서 출구까지 연결되어 있다고 가정한다.

제공 파일

✓ 제공된 소스폴더(LCP_PythonTest03WTest03_03)를 Open하여 알맞은 함수에 내용 구현

✓ 코딩하도록 지정한 부분 외에는 절대 수정하면 안됨

✓ 소스파일 내에서 data03_03.txt 파일을 사용하도록 코딩되어 있음.

※ 소스파일 Open 방법은 “실기 가이드” 참고

배점

아래 제시된 기능의 구현 완성도에 따라 점수를 부여함
(단, 실행 에러 발생 시 점수가 부여되지 않으며, 제시된 요구사항과 다를 경우 감점 발생)

1. 정확한 결과의 출력과 합리적인 문제 해결 처리 방식(12점)

2. 2차원 리스트 및 튜플 처리 기능(3점)

3. 반복문과 조건식의 올바른 적용 및 이미 지나온 길에 대한 처리 방식(3점)

실행 예시

----- 1 회 -----
방 상태
[1, 1, 2, 3, 0]
[0, 0, 0, 1, 0]
[0, 5, 1, 3, 0]
[0, 1, 0, 0, 0]
[0, 2, 3, 3, 1]

#이동 거리 : 13
#이동 경로 : (0, 0) -->(0, 1) -->(0, 2) -->(0, 3) -->(1, 3) -->(2, 3) -->(2, 2) -->(2, 1) -->(3, 1) -->(4, 1) -->(4, 2) -->(4, 3) -->(4, 4)
#모은 도토리 개수: 27

----- 2 회 -----
방 상태
[1, 2, 2, 1, 0, 0]
[0, 0, 0, 1, 0, 0]
[2, 1, 2, 2, 0, 0]
[1, 0, 0, 0, 0, 0]
[3, 3, 1, 0, 0, 0]
[0, 0, 4, 2, 1, 1]

#이동 거리 : 17
#이동 경로 : (0, 0) -->(0, 1) -->(0, 2) -->(0, 3) -->(1, 3) -->(2, 3) -->(2, 2) -->(2, 1) -->(2, 0) -->(3, 0) -->(4, 0) -->(4, 1) -->(4, 2) -->(5, 2) -->(5, 3) -->(5, 4) -->(5, 5)
#모은 도토리 개수: 30

수고하셨습니다.

