

[파이썬 트랙] 1회차 월말평가 - Python



시험 과목

✓ Python

시험 목적

✓ Python 프로그래밍 기본 문법에 대한 이해

시험 유의 사항

- 1) 성실하게 테스트에 임할 것 (부정 행위 적발 시 강력 조치)
- 2) 각 문제별 스켈레톤 코드가 제공되며 해당 파일을 수정하여 정답 코드를 작성할 것
- 3) 코드 유사도 판단 프로그램 기준, 부정 행위로 판단 될 시
0점 처리 및 학사 기준에 의거 조치
- 4) 채점은 문항에 제공된 예제 테스트 케이스와 별개의 테스트 케이스로 진행
- 5) 일부 문제에서 python 내장 함수(min, max, len, sum) 사용 불가
(문제에 명시되어 있으며 사용시 해당 문제 감점 처리)
- 6) 풀지 못한 문항이 있다면, 작성한 코드를 주석 처리하거나 삭제하여
문제를 풀기 전 상태로 복구 후 제출 (pass 키워드 복구)

시험 코드 작성 유의 사항

최종 제출 코드가 다음 항목에 해당하는 경우, 감점 혹은 0점 처리 될 수 있음

- 1) Syntax Error로 인한 코드 실행이 불가능한 경우
- 2) 주석 설명이 없거나 미흡한 경우
- 3) 출력 결과에 정답과 무관하거나 불필요한 내용이 있는 경우
- 4) 코드 수정 및 추가가 금지된 영역에 코드가 작성된 경우

[파이썬 트랙] 1회차 월말평가 - Python



시험 환경

- Visual studio code(이하 vscode)를 이용한다.

코드 실행

- 코드 실행은 반드시 터미널에서 python 실행 명령어로 진행한다.

```
Edwin@LECTURE MINGW64 /d/SSAFY  
$ python problem01.py
```

정답 제출 안내

제출 안내 사항 미 준수 시, 감점 혹은 0점 처리 될 수 있음

1) 압축 및 제출 파일 이름

- 지역0반_홍길동
ex) 서울1반_홍길동 / 부울경2반_김싸피

2) 압축 폴더 구조

- 서울_1반_김싸피 폴더를 압축하여 제출

```
서울_1반_김싸피/  
    problem01.py  
    problem02.py  
    problem03.py  
    ...  
    problem11.py
```

제출 마감 직전, 서버 요청이 집중될 수 있으므로, 미리 제출하는 것을 권장.

(마감 시간 이후 제출 불가)

[파이썬 트랙] 1회차 월말평가 - Python



문제 01 (problem01.py) | 15점

- ❖ 다음은 100km 속도제한구역에서 규정속도를 위반한 차량들의 속도를 리스트 형태로 정리한 값이다.
- ❖ 제시된 데이터를 참고하여 규정속도를 위반한 차량의 총 대수를 구하는 함수 `count_over_speed`를 작성하시오.
- ❖ **Python 내장 함수 len은 사용할 수 없다.**

```
[119, 124, 178, 192,]
```

문제 02 (problem02.py) | 10점

- ❖ 다음은 100km 속도제한구역에서 규정속도를 위반한 차량들의 속도를 리스트 형태로 정리한 값이다.
- ❖ 제시된 데이터를 참고하여 규정속도를 위반한 차량들의 평균 속도를 구하는 함수 `calc_average`를 작성하시오.
- ❖ **Python 내장 함수 len, sum은 사용할 수 없다.**

```
[119, 124, 178, 192,]
```

[파이썬 트랙] 1회차 월말평가 - Python



문제 03 (problem03.py) | 15점

- ❖ 물품의 상세 정보를 딕셔너리에 저장하고, 각 상세 정보 딕셔너리들을 물품 구매 리스트 SSAFANG_ITEM_LIST에 저장하였다.
- ❖ SSAFANG_ITEM_LIST에 담긴 모든 물품을 구매하는데 든 총 비용을 계산하는 함수 **calc_price**를 작성하시오. 각 물품별 구매 비용은 (price * count)로 계산한다.
- ❖ **Python 내장 함수 sum은 사용할 수 없다.**

```
SSAFANG_ITEM_LIST = [
    {
        'name': 'Galaxi Buds',
        'count': 3,
        'price': 150000,
    },
    ...
]
```

문제 04 (problem04.py) | 10점

- ❖ 파이썬 컨소시엄에서 암호 해독 퀴즈가 개최되었다.
- ❖ 참여자 김싸피는 security_code 리스트에 담긴 정수가 security_str의 index임을 알아차렸다.
- ❖ security_str 문자열에서 security_code의 각 정수를 index로 하는 문자들만 하나로 모아 합치는 함수 **make_answer**를 작성하시오.

```
security_str = 'kXeF50o1spkSMsuuoAiklFeqYz'
security_code = [4, 11, 17, 21, 24]
answer = 'SSAFY'
```

[파이썬 트랙] 1회차 월말평가 - Python



문제 05 (problem05.py) | 10점

- ❖ 아래 지문은 딕셔너리로 된 알파벳 암호표이다.
- ❖ 암호표를 활용하여 주어진 단어를 숫자로 이루어진 비밀번호 문자열로 변환하는 함수 **change_pwd**를 작성하시오.
- ❖ 예를 들어 'abc' 라는 문자열을 비밀번호로 변환시키면 '380' 이 된다.
- ❖ 공백 문자열 (' ') 에도 할당된 숫자가 있음에 유의한다.
- ❖ 비밀번호는 반드시 문자열 형태로 반환되어야 한다.

```
secret_code = {
    'a': 3,   'b': 8,   'c': 0,   'd': 1,
    'e': 7,   'f': 6,   'g': 7,   'h': 1,
    'i': 2,   'j': 5,   'k': 3,   'l': 6,
    'm': 5,   'n': 0,   'o': 7,   'p': 9,
    'q': 8,   'r': 2,   's': 4,   't': 9,
    'u': 6,   'v': 3,   'w': 2,   'x': 4,
    'y': 8,   'z': 1,   ' ': 0,
}
```

[파이썬 트랙] 1회차 월말평가 - Python



문제 06 (problem06.py) | 10점

- ❖ 문장의 각 알파벳을 일정한 양의 정수 n만큼 우측으로 밀어서 다른 알파벳으로 바꾸는 암호화 방식을 카이사르 암호(Caesar cipher)라고 한다.
- ❖ 예를 들어, 'a'를 3만큼 밀면 'a' -> 'b' -> 'c' -> 'd' 순으로 밀려 'd'가 되는 형식이다.
- ❖ 인자로 주어진 문자열을 양의 정수 n만큼 우측으로 밀어 완성된 카이사르 암호를 반환하는 함수 **caesar**를 참고 내용을 활용하여 작성하시오.
- ❖ 대문자는 항상 대문자로 소문자는 항상 소문자로 변환되어야 한다.
- ❖ 암호문은 알파벳으로만 이루어져 있으며, 그 외 문자로 변환 될 수 없다.

참고 내용

- .islower(), .isupper() 메서드를 사용하여 소문자, 대문자 여부를 확인한다.
- ord 함수는 인자로 넘긴 알파벳을 각 문자에 해당하는 정수로 변환한다.
- chr 함수는 인자로 넘긴 숫자를 각 숫자에 해당하는 문자열로 변환한다.
- 'a' ~ 'z' 는 각각 97 ~ 122, 'A' ~ 'Z'는 각각 65 ~ 90으로 변환되며, 반대의 경우도 동일하다.
- 만약 알파벳 범위를 벗어나면 다시 처음으로(소문자인 경우 'a', 대문자인 경우 'A') 돌아가서 남은 범위만큼 계산한다.

예) 문자 'x'를 3번 밀게 된다면 'a'가 된다. 'x'를 29번 밀어도 'a' 가 된다.

문자열 'Ny'를 3번 밀게 된다면 'Qb' 가 된다.

```
ord('A')          # => 65
ord('z')          # => 122
chr(65)           # => A
chr(122)          # => z
'a'.islower()     # => True
'A'.islower()     # => False
s1 = 'z'
s1.isupper()      # => False
s2 = 'Z'
s2.isupper()      # => True
```

[파이썬 트랙] 1회차 월말평가 - Python



문제 07 (problem07.py) | 10점

- ❖ 1보다 큰 자연수 중, 약수(나누어 떨어지는 수)가 1과 자기 자신 뿐인 수를 소수라고 한다.
- ❖ 양의 정수 n 이 인자로 주어질 때, n 미만인 소수의 합을 반환하는 함수 `sum_primes` 를 작성하시오.
- ❖ 단, 소수 17은 최종 합에 더하지 않고 다음 연산으로 건너뛴다.
- ❖ 주어지는 양의 정수 n 은 20 이상 100 미만 이다.
- ❖ **Python 내장 함수 `sum`은 사용할 수 없다.**

문제 08 (problem08.py) | 5점

- ❖ 주어진 문자열에서 정수로 변환 가능한 값들만 모아 모두 더하는 함수 `calculate_sum_number` 를 작성하시오.
- ❖ 문자열 'ab123c45d6' 이 인자로 주어지면, 이 중에서 숫자만 골라 더한 21을 반환한다.
- ❖ 문자가 숫자인지 확인하기 위해 문자열 메서드, `isdigit()`을사용한다.
- ❖ `isdigit()` 메서드는 문자열이 숫자로 구성되어 있다면 `True`를 반환한다.
- ❖ **Python 내장 함수 `sum`은 사용할 수 없다.**

```
'a'.isdigit()      # => False
'1'.isdigit()      # => True
'abc123'.isdigit() # => False
s1 = '123'
s1.isdigit()       # => True
s2 = 'ab12'
s2.isdigit()       # => False
```

[파이썬 트랙] 1회차 월말평가 - Python



문제 09 (problem09.py) | 5점

- ❖ 양의 정수 n 이 주어지면 n 의 각 자리 숫자들의 합을 반환하는 재귀 함수 `sum_digits` 를 작성하시오.
- ❖ n 이 123일 때, 반환 값은 각 자리 수 1, 2, 3의 합 6이 되어야 한다.
- ❖ 만약 n 이 한 자리 수인 경우, n 을 반환한다.
- ❖ 주어지는 n 의 자리 수는 1자리 이상 10자리 미만 이다.
- ❖ **재귀 함수를 구현하지 않은 경우 감점 처리된다.**
- ❖ **Python 내장 함수 `sum`은 사용할 수 없다.**

문제 10 (problem10.py) | 5점

- ❖ 주어진 리스트 `sample_matrix`는 3 X 3 정사각행렬로, 리스트에 다수의 리스트를 담아 표현한다.
- ❖ 주어진 리스트 `sample_matrix`의 값들 중, `sample_matrix[1][1]` 위치에는 1이 할당되어 있다.
- ❖ 이 때, 리스트 `sample_matrix`가 가진 1의 위치 좌표는 (1, 1)이 된다.
- ❖ 예제와 같이 $N \times N$ 정사각행렬에 1이 하나만 존재한다고 할 때, 1의 값이 할당된 위치 좌표를 반환하는 함수 `find_one`를 작성하시오.
- ❖ 반환 값의 순서는 (행 인덱스, 열 인덱스) 순서로 반환한다.
(`sample_matrix[행 인덱스][열 인덱스]`)
- ❖ 정사각행렬의 크기 N 은 $1 < N < 11$ 이다.
- ❖ **Python 내장함수 `len`, `min`, `max`, `sum`은 사용할 수 없다.**

```
sample_matrix = [
    [0, 0, 0],
    [0, 1, 0],
    [0, 0, 0]
]
```


[파이썬 트랙] 1회차 월말평가 - Python



문제 11 (problem11.py) | 5점

- ❖ 주어진 행렬의 행과 열의 합 중에서 가장 큰 값을 구하시오.
- ❖ 더 큰 합이 행의 합인 경우, 'row' 문자열을 합과 함께 반환하고, 열의 합인 경우, 'col' 문자열을 합과 같이 반환하시오.
- ❖ 결과는 반드시 ('row', 100) 혹은 ('col', 100)과 같이 튜플 형태로 반환한다.
(행/열 여부, 총 합)
- ❖ 행렬은 정사각행렬이 아닐 수 있다.
- ❖ 단, 빈 행렬이 주어지지 않는다.
- ❖ 행렬의 크기는 $1 < \text{row}, \text{col} < 10$ 이다.
- ❖ **Python 내장함수 len, sum, max는 사용할 수 없다.**

```
example_matrix = [    # 최대 합은 행의 합 58
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 10, 11, 12],
    [13, 14, 15, 16]
]

example_matrix2 = [   # 최대 합은 열의 합 102
    [11, 5, 49, 20],
    [28, 16, 20, 33],
    [63, 17, 1, 15]
]
```