

2016년도 2학기 컴퓨터프로그래밍기초(12729) 기말고사

학번: _____ 이름: _____

- 조급한 마음을 줄이고 한 줄 한 줄 천천히 읽어 가며 문제를 풀어 주세요.
- 시험은 90분 동안 진행되며 만점은 100점입니다.

1. 다음 중 Python의 함수에 대한 설명으로 알맞지 않은 것을 고르세요. (답 1개, 6점) ----- ()

- 1) def 키워드로 시작하는 '함수 정의'를 써서 정의합니다.
- 2) 함수의 실행을 중단할 때는 break문을 사용합니다.
- 3) 필요한 경우 인수를 통해 값을 받아 올 수 있습니다.
- 4) 자주 사용하는 기능을 함수로 정의해 두고 쓰면 편합니다.

2. 다음 중 Python의 list에 대한 설명으로 알맞지 않은 것을 고르세요. (답 2개, 6점) ----- ()

- 1) list는 형식 이름입니다.
- 2) list에 내용물이 몇 개 있는지를 알고 싶을 때는 `length()`를 사용합니다.
- 3) 특정 범위의 숫자 list를 새로 만들고 싶을 때는 `range()`를 사용합니다.
- 4) `numbers`라는 list의 첫 번째 칸에 있는 값을 가져올 때는 `numbers[1]`을 사용합니다.

3. 프로그래밍 전략들 중 특히 중요한 top-down approach에 대해 30자 내외로 설명해 주세요. (8점)

()

(4, 5) 아래 예제 코드를 읽고 물음에 답하세요.

```
def three():  
    return '3'  
  
def run():  
    print three()  
  
run()
```

4. 위 예제 코드에 함수 정의는 총 몇 개 들어 있나요? (6점) ----- ()

5. F5를 눌러 위 예제 코드를 실행했을 때의 결과를 한 문장으로 설명해 주세요. (4점)

()

- 이 문제들을 풀 때의 주의사항:
 - 함수 내용물이 아닌 다른 코드를 적으면 0점으로 간주됩니다!
 - 모든 함수는 10개 미만의 문장으로 완성할 수 있습니다.
 - 적다가 크게 실수해서 다시 적고 싶은 경우 아래에 있는 여분의 답안 칸을 이용해 주세요.
(이 때, 반드시 위 답안 칸의 내용에 가로줄을 그어 '안 쓰는 칸'임을 표기해 주세요. 그렇지 않은 경우 아래에 무엇을 적든 위 칸의 내용으로 채점을 수행합니다)
 - 문제에서 제시하는 함수/문장 외에도 온/오프라인 수업에서 다룬 다른 함수/문장을 쓸 수 있습니다.

- 사용할 수 있는 함수:
 - `write(text)` - 인수 `text`의 값을 출력합니다.
 - `read_key()` - 사용자가 누른 키 하나를 입력받아 해당 키 값을 `return`합니다.
- 사용할 수 있는 문장:
 - `if문(if부분, else부분)`

```
def are_you_sure(prompt):
```

```
def are_you_sure(prompt):
```

7. int 형식 값 하나를 사용자로부터 입력받고,

해당 값을 인수 data에 담긴 list의 맨 뒤에 추가하는 함수 append_number()를 완성해 주세요. (8점)

- 사용 가능한 함수 및 수식:

- `read_line()` – str 형식 값 하나를 사용자로부터 입력받아 return합니다.
- `int(x)` – x 값을 int 형식 값으로 바꾸는 수식입니다.
- `data.append(object)` – data에 담긴 list의 맨 뒤에 인수 object에 담긴 값을 추가합니다.

```
def append_number(data):
```

```
def append_number(data):
```

8. 아래 조건에 맞도록 함수 rank()를 완성해 주세요. (14점)

```
rank(data, val)
```

인수

data는 값들이 들어 있는 list입니다.

val은 순위를 구할 값입니다.

참고 문헌

1. 새 변수 `result`에 `int` 형식 값 1을 담으세요.
2. 반복문을 사용하여 `data`에 있는 값 하나하나를 꺼내고,
꺼낸 그 값이 `val`의 값보다 더 크면 `result` 값에 1을 더하세요(순위 하락).

return 값

변수 result에 담긴 값(방금 구한 순위 값)을 return합니다.

- 사용 가능한 문장 및 수식:

- for문(첫 줄은 for number in data: 와 같이 적음)

```
def rank(data, val):
```

```
def rank(data, val):
```

(9, 10, 11) 우리 시험의 남은 세 문제는 처음부터 직접 함수를 정의하는 문제입니다.

- 이 문제들을 풀 때의 주의사항:

- 모든 함수는 10개 이하의 문장으로 완성해야 합니다.

10개보다 많은 문장을 사용한 경우 초과한 수만큼 감점이 적용됩니다.

- ◆ 단, if문, for문, while문과 같은 복합적인 문장 자체는 세지 않습니다(간단한 문장만 셉니다).

- 적다가 크게 실수해서 다시 적고 싶은 경우 아래에 있는 여분의 답안 칸을 이용해 주세요.

(이 때, 반드시 위 답안 칸의 내용에 가로줄을 그어 '안 쓰는 칸'임을 표기해 주세요. 그렇지 않은 경우 아래에 무엇을 적든 위 칸의 내용으로 채점을 수행합니다)

- 문제에서 제시하는 함수/문장 외에도 온/오프라인 수업에서 다룬 다른 함수/문장을 쓸 수 있습니다.

- **(중요)** 만약 문제가 어려워서 답을 적을 수 없는 경우, Python 문장 대신 주석 (#로 시작하는 한 줄)을 써서 해당 위치에서 실행해야 할 문장의 의미를 한국어 / 영어로 적어 둘 수 있습니다. 이렇게 적은 내용이 타당한 경우, 해당 문장에 대한 부분점수의 최대 절반만큼을 획득할 수 있습니다.

- 그러니 마지막까지 힘 내 주세요!

9. 아래 조건에 맞도록 함수 `get_fare()`을 정의해 주세요. (13점)

```
get_fare(age)
```

인수: age는 지하철에 탑승하려는 사람의 나이 값입니다.

목표: 아래 표에 따라 해당 나이 값에 맞는 요금 값을 return해야 합니다:

구분(나이 범위)	요금(원)
성인(19 이상)	1250
청소년(13 이상, 19 미만)	720
어린이(13 미만)	450

- 사용 가능한 문장:

- if문(if부분, elif부분?, else부분?)

[illegible]A full-page sheet of white graph paper featuring a uniform grid of thin, light gray horizontal and vertical lines. The grid consists of small squares covering the entire area of the page.

10. 먼저, 아래 코드를 잘 읽어 주세요: (13점)

```
def process():
    dap = 'Enter'

    while True:
        # !!! 이동 - 출력
        move(0, 2)
        write('내 마음 속 키를 맞춰봐>')

        choice = read_key()

        if choice == dap:
            # !!! 이동 - 출력
            move(0, 0)
            write('맞혔네.')

            # !!! 이동 - 출력
            move(0, 1)
            write('이거 하나 맞혔다고 뿌듯해 하는건 아니겠지?')

            break
```

위 코드에서 주석으로 표기된 부분들은 모두 '커서 이동'을 위한 `move()`와 '현재 커서 위치에 출력'을 위한 `write()`로 구성되어 있습니다. 매 번 이렇게 두 줄씩 적는 것은 번거로우므로, 이 두 작업을 한 번에 수행하는 새로운 함수를 만들어 사용하려고 합니다. 이러한 조건에 맞는 함수 `write_at()`을 아래에 정의해 주세요.

주의: `write_at()`은 두 작업을 수행하기 위한 적절한 인수를 가지고 있어야 합니다!

[illegible]A large grid of graph paper with 20 columns and 10 rows. The grid is composed of small squares, with a slightly larger square at the top left corner, likely for a title or header. The grid is intended for drawing a graph.

11. 아래 코드는 '3초 안에 Enter를 눌러야 이기는 게임' 코드의 일부입니다: (14점)

```
def do_the_game():
    dap = 'Enter'
    timeout = 3.0

    # time.clock()은 현재 시각에 해당하는 값(단위는 초)을 return하는 함수입니다.
    time_start = time.clock()

    while True:
        # process_input(dap)은,
        # 사용자가 키를 눌렀고 해당 키가 dap과 같으면 True를 return하는 함수입니다.
        # (사용자가 키를 안 눌렀거나, dap이 아닌 키를 누르면 False를 return합니다)
        if process_input(dap) == True:
            write('You won!')
            break

        # !!! 이 부분에 주목해 주세요!
        if check_time_over(time_start, timeout) == True:
            write('You lose...')
            break
```

주석으로 표시된 부분에서 사용되고 있는 함수 `check_time_over()`가 프로그램의 목표 및 주변 문맥에 맞게 잘 실행되도록 아래에 정의해 주세요. (부분점수 많아요)

진행 순서:

- 먼저, 게임의 이름과 각 변수들의 이름에 주목하면서 코드를 꼼꼼히 읽어 보세요.
- `check_time_over()`가 언제 이 함수가 True를 return해야 할 지 생각해 보세요.
- 위 예제에서 사용한 `time.clock()`을 활용하여 `check_time_over()`의 내용물을 채워 주세요.

A full page of blank graph paper with a uniform grid of small squares. The grid consists of 20 columns and 15 rows, providing a structured area for drawing or writing.

