

2023년도 2학기 “소프트웨어적 사고” 중간고사 라이브코딩

이름 :

학번 :

■ 중간고사 라이브코딩 관련 공지사항

- 총 130 점
- 시간 : 2023. 10. 26(목) 13:30 ~ 14:55
- 장소 : 미래관 445
- Off-line / Open book

- 제출
 - o 작성된 프로그램과 캡처한 실행화면을 ecampus로 중간고사 제출 link로 upload하기 바랍니다.
 - o 파일명을 반드시 다음과 같이 사용하십시오. (반드시 Idle 환경에서 통합실행 가능해야 합니다.)
프로그램 화일명 : mid-학번-이름.py
실행화면 캡처화일 : mid-학번-이름.png
 - o 종료시간 이후에는 형평성을 위해 프로그램 upload 절대 불가 (ecampus에 upload시간 setting). 시험 이후 이메일 등 개인 제출 절대불가 (제출해도 평가하지 않음).

- 채점 기준
 - o **Error 발생하여 실행이 안되면 무조건 0점.**
 - o Error 없이 실행이 되는 경우에만 문제에서 제시한 논리 및 조건 (공통조건 포함)에 따라 채점 진행

- 주의사항
 - o Cheating 절대금지 : 아래의 경우 본 과목 F로 처리
 - 1) 남의 프로그램을 표절하거나 다른 사람이나 인공지능 tool에 위탁하여 코딩을 하는 행위. (국민대학교 관련교칙에 따라 엄격하게 처리합니다.)
 - 2) 제시된 logic을 이용하여 문제를 해결하지 않고 출력문 등을 사용하여 프로그램 입출력형식만을 보여주는 fake 행위 등 편법을 사용하는 것.
 - 3) 일반적인 모든 데이터가 아닌 몇 가지 예제 데이터 만으로 한정지어 성공/실패를 확인하도록 코딩을 하는 행위 (hard coding)
 - 4) 이외 학생 및 미래의 전문직 엔지니어로서 양심에 반하는 행위 등

- 다음의 선서문을 제출하는 프로그램의 첫 번째 라인에 주석문으로 올리기 바랍니다.

'''

선서 : 나는 양심에 따라 절대 부정행위를 하지 않겠습니다. 부정행위 적발 시 모든 민사, 형사상의 법적책임을 지겠습니다. 이름 홍길동 (20233456), 2023년 10월 26일 (목)

'''

[2023년도 2학기 “소프트웨어적 사고” 중간고사 live coding 문제지]

■ 공통조건

- Comment를 충분히 넣으시오.

문제의 주어진 단계를 comment로 넣으시오. 필요시 변수, 논리 등을 설명하는 comment를 넣으시오.

- Echo checking을 반드시 하시오.
- Debugging 기법을 활용하시오. (if D)
- 다음의 프로그램 입출력형식을 따르시오.

```
IDLE Shell 3.11.5
File Edit Shell Debug Options Window Help

>> 5개의 값을 한 개씩 차례대로 입력하세요.

>> in_list에 추가할 1번째 값을 입력하세요 : Hi, Everybody !
>>> 입력값(str type) : Hi, Everybody !

>> WARNING : 입력값이 숫자형(integer, floating-point, hexadecimal)이 아닙니다.

>> in_list에 추가할 1번째 값을 숫자형으로 다시 입력하세요 : 12
>>> 입력값 : 12
>>> 입력값 12가(이) integer 12(으)로 type-casting되었습니다

>> in_list에 추가할 2번째 값을 입력하세요 : FG
>>> 입력값(str type) : FG

>> WARNING : 입력값이 숫자형(integer, floating-point, hexadecimal)이 아닙니다.

>> in_list에 추가할 2번째 값을 숫자형으로 다시 입력하세요 : FA
>>> 입력값 : FA
>>> 입력값 FA가(이) Decimal 250(으)로 type-casting되었습니다

>> in_list에 추가할 3번째 값을 입력하세요 : 18.99
>>> 입력값(str type) : 18.99
>>> 18.990000가(이) integer 18(으)로 type-casting되었습니다

>> in_list에 추가할 4번째 값을 입력하세요 : 23.88
>>> 입력값(str type) : 23.88
>>> 23.880000가(이) integer 23(으)로 type-casting되었습니다

>> in_list에 추가할 5번째 값을 입력하세요 : TT
>>> 입력값(str type) : TT

>> WARNING : 입력값이 숫자형(integer, floating-point, hexadecimal)이 아닙니다.

>> in_list에 추가할 5번째 값을 숫자형으로 다시 입력하세요 : 250
>>> 입력값 : 250
>>> 입력값 250가(이) integer 250(으)로 type-casting되었습니다

>> in_list : [12, 250, 18, 23, 250]

>> 오름차순 sort : [12, 18, 23, 250, 250]

>> min_max : {'min_key': 12, 'max_key': 250}

>> 최대값 250와 최소값 12의 gcd : 2

>> min_max : {'min_key': 12, 'max_key': 250, 'gcd': 2}
>>>
```

Ln: 52 Col: 0

■ 문제

5개의 입력값을 받아들이며 data type을 확인한 후 list에 저장한다. 이 값 들 중 최대, 최소 2개의 값만을 선택하여 이 두 값의 gcd를 구한다.

* 수업자료 4.1example2-1.py, identify_haxa.py 참조

1) input()을 이용하여 총 5개의 string 값을 한 개씩 받아들이며 변수에 저장한다. 그리고 이 값들을 저장할 list data type 변수를 생성하시오(변수명 in_list).

>> 5개의 값을 한 개씩 차례대로 입력하세요.

>> in_list에 추가할 1번째 값을 입력하세요 :

1-1) 하나 씩 받아들인 값이 숫자(integer, floating-point, hexa-decimal 중 하나)가 아닌 경우 잘못된 입력임을 밝히고 숫자를 받아들일 때까지 계속 입력을 받도록 한다.

>> in_list에 추가할 1번째 값을 입력하세요 : Hi, Everybody !

>>> 입력값(str type) : Hi, Everybody !

>> WARNING : 입력값이 숫자형(integer, floating-point, hexadecimal)이 아닙니다.

>> in_list에 추가할 1번째 값을 숫자형으로 다시 입력하세요 :

1-2) 1-1)단계 에서 성공적으로 받아들인 숫자가

1-2-1) integer인 경우 type casting을 거쳐 append()를 이용하여 in_list에 저장하시오.

>> in_list에 추가할 1번째 값을 숫자형으로 다시 입력하세요 : 12

>>> 입력값 : 12

>>> 입력값 12가(이) integer 12(으)로 type-casting되었습니다

1-2-2) hexadecimal여부 확인

수업시간에 배운 identify_hexa.py 코드를 이용하여 hexa-decimal인지 검증한 후 hexa-decimal이 맞는 경우에만 append()를 이용하여 in_list에 저장하시오.(10진수로 저장하여도 좋음)

>> in_list에 추가할 2번째 값을 입력하세요 : FG

>>> 입력값(str type) : FG

>> WARNING : 입력값이 숫자형(integer, floating-point, hexadecimal)이 아닙니다.

>> in_list에 추가할 2번째 값을 숫자형으로 다시 입력하세요 : FA

>>> 입력값 : FA

>>> 입력값 FA가(이) Decimal 250(으)로 type-casting되었습니다

1-2-3) floating-point인 경우 integer로 type casting한 후 append()를 이용하여 in_list에 저장하시오.

>> in_list에 추가할 3번째 값을 입력하세요 : 18.99

>>> 입력값(str type) : 18.99

>>> 18.990000가(이) integer 18(으)로 type-casting되었습니다

1-2-4) 위의 경우가 모두 아닌 경우 다시 값을 받아들인다.

```
>> in_list에 추가할 5번째 값을 입력하세요 : TT
>>> 입력값(str type) : TT
>> WARNING : 입력값이 숫자형(integer, floating-point, hexadecimal)이 아닙니다.
>> in_list에 추가할 5번째 값을 숫자형으로 다시 입력하세요 :
```

1-3) 5개의 값을 받아들일 때까지 step 1-1) ~ step 1-2)를 계속한다.

1-4) 5개의 값을 받아들인 후 최종 생성된 in_list를 출력한다.

```
>> in_list : [12, 250, 18, 23, 99]
```

1-5) deep copy를 이용하여 오름차순으로 sort한 후 출력하시오.

```
>> 오름차순 sort : [12, 18, 23, 99, 250]
```

2)

2-1) in_list에 입력된 5개의 값 중 제일 큰 수(large)와 제일 작은 수(small)을 찾으시오.

2-2) dictionary data type 다음과 같이 저장 후 출력하시오.

dictionary 변수명 : min_max

key : "min_key", "max_key"

value : 2-1)에서 얻은 값

```
>> min_max : {'min_key': 12, 'max_key': 250}
```

3)

3-1) 2-1)단계에서 생성한 min_max에서 제일 큰 수(large)와 제일 작은 수(small)를 dictionary key로 값을 읽어와서 최대공약수(GCD)를 구하여 변수 gcd에 저장하고 출력하시오.

```
>> 최대값 250와 최소값 12의 gcd : 2
```

3-2) dictionary min_max에서 다음과 같이 저장한 후 출력하시오.

key : "gcd"

value : 3-1)에서 얻은 값

```
>> min_max : {'min_key': 12, 'max_key': 250, 'gcd': 2}
```

- END -