

#1

카테고리

- 변수
- 조건문

문제

학생의 시험 점수를 받아 학점을 출력하는 프로그램을 만들어 봅시다.

학생의 시험 점수를 물어보고 시험 점수가 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 60점 이상이면 D, 그 이외의 점수이면 F를 학점으로 주는 프로그램을 작성해 보세요.

- 시험 점수는 정수형(integer, int)으로 입력 받습니다.
- 출력은 아래 중 하나로 출력합니다.
 - A학점입니다.
 - B학점입니다.
 - C학점입니다.
 - D학점입니다.
 - F학점입니다.

입출력 예시

예시 1번

```
성적을 입력하시오: 90
-----
A학점입니다.
```

- 90점을 입력 받고 “A학점입니다.”로 출력합니다.

#2

카테고리

- 반복문
- 재귀 연산

문제

반복문을 통해 피보나치 수열을 구현해 봅시다.

피보나치 수 : 첫째 및 둘째 항이 1이며, 그 뒤의 항은 바로 앞 두 항의 합이 되는 수열이다.

N을 입력 받아 F(1)부터 F(N)까지 공백을 기준으로 출력하는 프로그램을 작성해 봅시다.

$$F(N) = F(N - 1) + F(N - 2) \quad (N \geq 3)$$
$$F(1) = F(2) = 1$$

- 입력값 N은 정수형(integer, int)으로 입력 받습니다.

참고 사항) 피보나치 수열의 전개



1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...

- 첫 항과 두 번째 항은 1
- 세 번째 항은 바로 앞 두 항의 합으로 $1+1=2$
- 네 번째 항은 바로 앞 두 항의 합 $1+2=3$
- 다섯 번째 항은 바로 앞 두 항의 합 $2+3=5$

입출력 예시

```
숫자를 입력하시오: 10
-----
1 1 2 3 5 8 13 21 34 55
```

- 10을 입력 받고 F(1)부터 F(10)의 값을 공백을 두고 출력합니다.

#3

카테고리

- 파이썬의 자료형
 - 자료형에 따른 적절한 반복문
-

문제

비품이 개수가 3개 이하로 줄어들지 않게 비품의 재고를 관리하는 재고품 관리 프로그램을 만들어봅시다.

재고품 관리 프로그램의 작동 방식은 아래와 같습니다.

- 물건의 가짓수를 정수형 자료형(integer, int)으로 입력받습니다.
- 물건의 가짓수만큼, 아래와 같은 정보들을 입력받습니다.
 - 품명(문자열 자료형, string, str)
 - 가격(정수형 자료형, integer, int)
 - 개수(정수형 자료형, integer, int)
- 해당 정보들을 입력받은 뒤, 입력받은 개수가 3개 미만인 물건들을 3개까지 채워넣습니다.
- 그리고 아래의 정보들을 출력합니다.
 - 어떤 물건을 샀는가
 - 해당 물건을 몇 개나 샀는가
 - 물건을 구매하는 데 총 얼마의 비용이 추가 지출되었는가

입출력예시

예시 1번

입/출력

✓ 물건의 가짓수 입력: 2

--1 번째 물건 정보 입력 시작--

품명: 커피

가격: 200

개수: 1

--1 번째 물건 정보 입력 종료--

--2 번째 물건 정보 입력 시작--

품명: 우유

가격: 500

개수: 0

--2 번째 물건 정보 입력 종료--

✓ 물건 재고 정보 입력 완료

✓ 추가 구매가 필요한 물건들과 갯수

- 커피: 2개

- 우유: 3개

✓ 총 금액: 1900

설명

- 물건의 가짓수를 2개로 입력 받았기에, 아래와 같은 항목들을 2번 입력 받습니다.
 - 품명
 - 가격
 - 갯수
- 커피가 3개가 되려면 2개를 더 구매해야 하고, 우유가 3개가 되려면 3개를 더 구매해야 합니다.
- 그래서, 추가로 구매해야 할 물건과 그 수량을 아래와 같이 출력합니다.
 - 커피: 2개
 - 우유: 3개
- 그리고, 커피 2개를 추가 구매해야 하니 400, 우유 3개를 구매해야 하니 1500원, 다 합쳐서 추가로 지출해야 하는 총 금액인 1900(원) 또한 다음과 같이 출력합니다.

#4

카테고리

- 함수와 모듈
- 문자열 다루기

—

문제

- 두 개의 날짜를 월/일 로 입력받아, 두 날짜 사이의 날짜 수를 계산하는 프로그램을 만듭니다.

요구사항

- 아래의 두 함수를 만들고, 두 함수를 사용하여 위의 문제를 해결해야 합니다.
- `get_days_of_month(month: int) -> int:`
 - 정수형 자료형(int, integer)으로 월을 입력받습니다.
 - 해당 월의 날짜 수를 반환합니다.
- `count_days(from: str, to: str) -> int:`
 - 월/일 형식으로 이루어진 문자열(str, string) 자료형으로
 - 시작날짜인 from
 - 종료날짜인 to 를 받습니다.
 - 두 날짜 사이의 간격이 며칠인지 반환합니다.

추가설명

- 2월은 28일 / 1, 3, 5, 7, 8, 10, 12 월은 31일 / 그 외의 월은 30일로 간주합니다.
- 년도 및 년도에 따라 달라질 수 있는 요소들(윤달/윤년)은 고려하지 않습니다.
- 1자리 숫자는 0X 로 입력하지 않고, 그냥 X라고 입력합니다.
 - 예를 들어, 3월 3일을 입력하고 싶은 경우 '03/03' 이 아닌 '3/3' 을 입력합니다.
- 시작일은 포함하지 않고, 종료일만을 포함합니다.
 - 예를 들어, 3월 15일과 3월 16일 사이의 날짜는 2일이 아닌 1일입니다.
- 종료 날짜는 시작 날짜보다 앞설 수 없다는 상황만을 고려합니다.
 - 그렇기에, 앞서는 경우 예외처리 또한 고려하지 않습니다.

입출력예시

예시 1번

입/출력

시작 월/일을 month/date 형식으로 입력하세요: 10/9
종료 월/일을 month/date 형식으로 입력하세요: 10/10

10/9 부터 10/10 까지는 1 일입니다.

설명

- 10월 9일부터 10월 10일까지는 시작일을 포함하지 않는 경우 1일이 존재합니다.

예시 2번

입/출력

시작 월/일을 month/date 형식으로 입력하세요: 3/1
종료 월/일을 month/date 형식으로 입력하세요: 4/30

3/1 부터 4/30 까지는 60 일입니다.

설명

- 3월 1일부터 4월 30일까지는 시작일을 포함시키지 않는 경우 60일이 존재합니다.
 - 3월 1일
 - 1월: 31일 / 2월: 28일 / 3월: 1일 -> 총 60일
 - 4월 30일
 - 1월: 31일 / 2월: 28일 / 3월: 31일 / 4월: 30일 -> 총 120일
 - 120일 - 60일 -> 60일