

## [문제-1] 두 컵에 담긴 물 교환

- 컵 두 개(a, b)에 각각 음료가 담겨있다. (a에는 파란색 물이 b에는 빨간색 물이)
- 빈 컵(t)은 얼마든지 사용할 수 있다.

[질문] 어떤 과정(단계)을 거쳐야 두 컵에 물을 교환할 수 있는지 적어보세요.

```
In [3]: ## [문제-1] 두 컵에 담긴 물 교환
a = 'Red'
b = 'Blue'
temp = ''

print(">Before: ", a, b)
temp = a
a = b
b = temp
print(">After: ", a, b)
```

```
>Before: Red Blue
>After: Blue Red
```

## [문제-2] 할인율 적용 순서

- 할인율 적용 대상은 아래와 같다.

[할인율-a] “지역주민”은 30% 할인 대상이다.

[할인율-b] “국가유공자/현역군인”은 50% 할인 대상이다.

[할인율-c] “70세 이상” 어르신은 100% 할인 대상이다.

[질문] 할인율을 최대한 높게 결정해주어야 한다면, 할인율 적용을 위해 어떤 순서로 질문을 하여야 가장 적게 질문할 수 있는지 그 순서를 적어보세요.

```
In [6]: ## [문제-2] 할인율 적용 순서
fee = 10000 #입장료
print(">입장료는 %d원입니다." %fee)
print(">신분이 어떻게 되는지 y나 기타 키로 답하세요.")

status = input(">>70세 이상이신가요(y)? ").strip().lower()
if status == 'y':
    rate = 100
else:
    status = input(">>국가유공자나 현역군인이신가요(y)? ").strip().lower()
    if status == 'y':
        rate = 70
    else:
        status = input(">>지역주민이신가요(y)? ").strip().lower()
        if status == 'y':
            rate = 30
```

```

        else:
            rate = 0

#fee = fee * (1 - rate/100)
fee = fee - (fee * rate / 100)
print(">>>%d%%가 할인되어 %d원입니다." % (rate, fee))

```

>입장료는 10000원입니다.  
>신분이 어떻게 되는지 y나 기타 키로 답하세요.  
>>>100%가 할인되어 0원입니다.

## [문제-3] 할인 적용 여부 판단 조건

○ “할인” 조건은 다음과 같다.

- “국가유공자”인 경우 또는 “현역 군인”인 경우

[질문] “비할인” 조건을 설명하여보세요.

```

In [11]: ## [문제-3] 할인 적용 여부 판단 조건
print(">신분이 어떻게 되는지 y나 기타 키로 답하세요.")

status1 = input(">>국가유공자이신가요(y)? ").strip().lower()
status2 = input(">>현역군인이신가요(y)? ").strip().lower()
if (status1 != 'y') and (status2 != 'y'):
    print(">>>할인 대상이 아닙니다.")
else:
    print(">>>할인 대상입니다.")

```

>신분이 어떻게 되는지 y나 기타 키로 답하세요.  
>>>할인 대상이 아닙니다.

## [문제-4] “합격”과 “불합격” 판단

○ “합격” 조건은 다음과 같이 조건-A와 조건-B를 모두 만족시켜야 한다.

- [조건-A] 각 과목에서 40점 미만이 없어야 한다.

- [조건-B] 모든 과목의 평균 점수가 60점 이상이어야 한다.

[질문] “불합격” 조건을 설명하여보세요.

```

In [20]: ### [문제-4] “합격”과 “불합격” 판단
cnt_subject = 0      #과목 수
cnt_blow = 0         #과락 점수 개수
tot_score = 0        #총점
in_score = 0          #과목 점수 입력 값

## 과목별 점수 입력
while in_score >= 0:
    in_str = input(">과목 점수 입력(Enter는 종료):")  #과목 점수 입력
    if len(in_str) == 0: #입력 종료(빈 값 입력 시)
        break
    in_score = int(in_str) #입력 점수 숫자로 변환
    cnt_subject = cnt_subject + 1

```

```

        tot_score = tot_score + in_score
        if in_score < 40: #과락 점검
            cnt_blow += 1

## 과락 및 평균으로 합/불 판단
if cnt_subject > 0:
    avg = tot_score / cnt_subject
    if (cnt_blow == 0) and (avg >= 60): #합격 판단
        print(">>합격입니다.! ")
        print("Average: %4.1f" %avg)
    else:
        print(">>아쉽지만, 불합격입니다.! ")
        print(">>> Average: %4.1f *과락과목: %d개" %(avg, cnt_blow))

```

>>아쉽지만, 불합격입니다.!  
>>> Average: 50.0 \*과락과목: 1개

## [문제-5] 정n각형 그리기

○ 아래와 같은 작업을 4번 반복하면 정4각형 괘적을 그리고 제 자리에 돌아올 수 있다.

(i) 앞으로 10보(step) 이동한다. (ii) 좌로 90도(angle) 회전한다.

**[질문]** 정n각형을 그리기 위해서 좌로 몇 도를 회전해야 하는지에 대한 일반화된 식을 적어보시오. (정확하게 n번 만에 제자리로 돌아오기 위한 회전 각도(angle) 산출 공식)

```
In [29]: ## [문제-5] 정n각형 그리기
import turtle as t
t.shape('turtle') #Turtle 모양

n = int(input(">몇 각형(3~n)? ")) #키보드 입력 처리
step = 100
angle = 360 // n # //는 나눈 둘

for _ in range(n): #n회 반복
    t.forward(step) #앞으로 100 pixcel 이동
    t.left(angle) #좌로 90도 회전

t.exitonclick() # 실행 창을 닫지 않도록
```

## [문제-6] 요일 알아내기

○ 오늘은 일요일이다.

**[질문]** 오늘로부터 1000번째 일수가 경과 한 날의 요일은 무슨 요일일까?

```
In [25]: ## [문제-6] 요일 알아내기
day_name = ['일', '월', '화', '수', '목', '금', '토']

to_day = '일' #기준일 요일
day_seq = day_name.index(to_day) #기준일(to_day)의 순서값 구하기
print(">오늘은 %s요일(%d)" %(day_name[day_seq], day_seq))
```

```

n = int(input("경과일 수 입력:")) #경과일 수 입력
day_seq = (day_seq + n) % 7 # n일 경과 후의 요일 순서값 구하기
print(">> %d일 후는 %s요일(%d)" %(n, day_name[day_seq], day_seq))

```

>오늘은 일요일(0)

>> 1000일 후는 토요일(6)

## [문제-7] 요일 순서 바꾸기

○ 아래 () 안은 각 요일에 대한 순서 **Index** 값이다.

>요일: 월(0), 화(1), 수(2), 목(3), 금(4), 토(5), 일(6)

>> **Original Index**는 월/화/수/목/금/토/일 순이다.

**[질문]** **Original Index** 값으로부터 **New Index** 값을 환산하는 공식을 적으시오.

>> **New Index**는 일/월/화/수/목/금/토 순이다.

```

In [31]: ## [문제-7] 요일 순서 바꾸기
import calendar
day_name = ['Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa', 'Su']
day_alpha = [0, 6, 5, 4, 3, 2, 1] #시작 요일 조정 값
start_day = 'Mo' #시작 요일

alpha = day_alpha[day_name.index(start_day)] #시작 요일 번호 조정 값 구하기

for i in range(7):
    x = (i+alpha) % 7 #조정된 요일 순서 값
    print("%s(%d)" %(day_name[i], x), end=' ')

```

Mo(0) Tu(1) We(2) Th(3) Fr(4) Sa(5) Su(6)

```

In [32]: ## [문제-7] 요일 순서 바꾸어 달력 출력
import calendar
day_name = ['Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa', 'Su']
start_day = 'Su' #시작 요일

## 시작 요일 조정
if start_day == 'Su':
    for i in range(6, 0, -1): #요일 순서 조정
        day_name[i] = day_name[i-1]
    day_name[0] = 'Su'

##[함수] 년,월을 받아 월의 마지막 일자 반환
def get_month_end(y, m):
    val = calendar.monthrange(y, m)
    return val[1]

##[함수] 년,월을 받아 1일의 요일 순서번호 반환(0이 월)
def get_month_weekday(y, m):
    val = calendar.monthrange(y, m)
    return val[0]

year = int(input("년도(4자리): "))
month = int(input("월(2자리): "))

wday = get_month_weekday(year, month) #요일 순서 번호 얻기

```

```

mend = get_month_end(year, month)      #월의 마지막 날 얻기
## 시작 요일에 따른 시작 번호 조정
if start_day == 'Su':
    wday = (wday + 1) % 7   #0을 일요일로 치환

##타이틀 출력
print("Wt%4d년 %2d월 WtWt" %(year, month))
for i in range(7):
    print(" %s" %day_name[i], end=' ')
print()
##일자 출력(월화수목금토일 순)
for _ in range(wday):  #시작일자만큼 둘여쓰기
    print(" ", end=' ')
wseq = wday
for dd in range(1, mend+1):
    if wseq >= 7: #7번째 일마다 줄 바꿈
        print()
        wseq = 0
    print("%4d" %dd, end=' ')
    wseq += 1

```

2023년 6월

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

## [문제-8] 거스름 돈 지불

○거스름 돈 135,000원을 지폐(5만/1만/5천/1천 원권)로 나누어 지불하여야 한다.

[질문] 큰 지폐 순으로 지급하여 지폐의 수를 최소화하려고 할 때 다음 질문에 대한 식을 적어보시오.

(연산자는 사칙연산(+-\*%), 나눈 몫(%), 나눈 나머지(/) 등을 사용)

<질문-A> 5만 원권의 개수를 구하는 식을 적어보시오.

<질문-B> 1만 원권의 개수를 구하는 식을 적어보시오.

```

In [33]: ## [문제-8] 거스름 돈 지불
## 5만원권 처리 > 1만원권 처리
remain = int(input("반환할 금액 입력(원)? ")) #남은 액수

coin_50000 = remain // 50000 #내준 개수: 50000으로 나눈 몫
print("> 50000원권: %d" %coin_50000)
remain = remain - (coin_50000 * 50000) #남은 액수: 50000으로 나눈 나머지
print(">> 남은 액수: %d" %remain)

coin_10000 = remain // 10000 #내준 개수: 10000으로 나눈 몫
print("> 10000원권: %d" %coin_10000)
remain = remain - (coin_10000 * 10000) #남은 액수: 10000으로 나눈 나머지
print(">> 남은 액수: %d" %remain)

```

```
> 50000원권: 2  
>> 남은 액수: 35000  
> 10000원권: 3  
>> 남은 액수: 5000
```

```
In [48]: ## [문제-8] 거스름 돈 지불  
## 최소 화폐 개수로 금액 반환  
unit = [50000, 10000, 5000, 1000, 500, 100, 50, 10, 5, 1] #List형 데이터 구조  
  
cnt_won = len(unit) #unit의 항목 개수 구하기  
remain = int(input("반환할 금액 입력(원)? ")) #남은 액수  
  
for i in range(cnt_won): # (0에서부터 cnt_won이 되기 전까지) 값을 i로 반복하며 반복  
    result = remain // unit[i] #내준 개수  
    if result != 0:  
        print("> %d원권: %d" %(unit[i], result))  
    remain = remain % unit[i] #남은 액수
```

  

```
> 50000원권: 2  
> 10000원권: 3  
> 5000원권: 1
```

## [문제-9] 행렬 값의 순서

○ 아래  $3 \times 4$  행렬에서 값 6(1행, 2열 값)은 시작 값(0번째 0)을 기준으로 6번째에 위치한다.

```
arr = [[0, 1, 2, 3], [4, 5, 6, 7], [8, 9, 10, 11]]
```

[질문-1] 만약  $15 \times 10$  행렬이라면 7행 8열의 값은 몇 번째 값에 해당하는가?

[질문-2] 만약  $15 \times 10$  행렬이라면 96번째 값은 몇 행 몇 열에 있는가?

```
In [51]: ## [문제-9] 행렬 값의 순서  
## [질문-1] 만약  $15 \times 10$  행렬이라면 7행 8열의 값은 몇 번째 값에 해당하는가?  
row = 15  
col = 10  
  
# 배열 순서번호로 초기화  
arr = [[col*i+j for j in range(col)] for i in range(row)] #2차원 배열 생성  
for r in arr: #2차원 배열 출력  
    print(r, end=' ')  
  
## <a> 식으로 순서 번호 환산  
r = 7 #행  
c = 8 #열  
seq_no = col * r + c  
print("Wn<a> %d행 %d열 값은 %d번째 값" %(r, c, seq_no))  
  
## <b> 찾아가는 방식으로 순서 번호 환산  
seq_no = -1 #순서 번호(0부터 시작) 초기값  
chk = ''  
for i in range(row):  
    for j in range(col):  
        seq_no = seq_no + 1  
        if i == r and j == c: #7행 8열  
            chk = 'break'  
    if chk == 'break':  
        break
```

```

        break #반복문 탈출
    if chk == 'break':
        break #반복문 탈출
print("b> %d행 %d열 값은 %d번째 값" %(r, c, seq_no))

```

```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9] [10, 11, 12, 13, 14, 15, 16, 17, 18, 19] [20, 21, 22,
23, 24, 25, 26, 27, 28, 29] [30, 31, 32, 33, 34, 35, 36, 37, 38, 39] [40, 41, 42, 4
3, 44, 45, 46, 47, 48, 49] [50, 51, 52, 53, 54, 55, 56, 57, 58, 59] [60, 61, 62, 63,
64, 65, 66, 67, 68, 69] [70, 71, 72, 73, 74, 75, 76, 77, 78, 79] [80, 81, 82, 83, 8
4, 85, 86, 87, 88, 89] [90, 91, 92, 93, 94, 95, 96, 97, 98, 99] [100, 101, 102, 103,
104, 105, 106, 107, 108, 109] [110, 111, 112, 113, 114, 115, 116, 117, 118, 119] [12
0, 121, 122, 123, 124, 125, 126, 127, 128, 129] [130, 131, 132, 133, 134, 135, 136,
137, 138, 139] [140, 141, 142, 143, 144, 145, 146, 147, 148, 149]
a> 7행 8열 값은 78번째 값
b> 7행 8열 값은 78번째 값

```

In [53]: ## [문제-9] 행렬 값의 순서

```

## [질문-2] 만약 15x10 행렬이라면 96번째 값은 몇 행 몇 열에 있는가?
row = 15
col = 10

# 배열 순서번호로 초기화
arr = [[col*i+j for j in range(col)] for i in range(row)] #2차원 배열 생성
for r in arr: #2차원 배열 출력
    print(r, end=' ')

```

## <a> 식으로 행, 열 값 환산

```

seq_no = 96
r = seq_no // col      #행 계산
c = seq_no % col      #열 계산
print("a> %d번째 값은 %d행 %d열" %(seq_no, r, c))

```

## <b> 찾아가는 방식으로 행, 열 값 환산

```

seq = -1 #순서 번호(0부터 시작) 초기값
chk = ''
for i in range(row):
    for j in range(col):
        seq = seq + 1
        if seq == seq_no: #96번째
            chk = 'break'
            break #반복문 탈출
    if chk == 'break':
        break #반복문 탈출
print("b> %d번째 값은 %d행 %d열" %(seq_no, i, j))

```

```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9] [10, 11, 12, 13, 14, 15, 16, 17, 18, 19] [20, 21, 22,
23, 24, 25, 26, 27, 28, 29] [30, 31, 32, 33, 34, 35, 36, 37, 38, 39] [40, 41, 42, 4
3, 44, 45, 46, 47, 48, 49] [50, 51, 52, 53, 54, 55, 56, 57, 58, 59] [60, 61, 62, 63,
64, 65, 66, 67, 68, 69] [70, 71, 72, 73, 74, 75, 76, 77, 78, 79] [80, 81, 82, 83, 8
4, 85, 86, 87, 88, 89] [90, 91, 92, 93, 94, 95, 96, 97, 98, 99] [100, 101, 102, 103,
104, 105, 106, 107, 108, 109] [110, 111, 112, 113, 114, 115, 116, 117, 118, 119] [12
0, 121, 122, 123, 124, 125, 126, 127, 128, 129] [130, 131, 132, 133, 134, 135, 136,
137, 138, 139] [140, 141, 142, 143, 144, 145, 146, 147, 148, 149]
a> 96번째 값은 9행 6열
b> 96번째 값은 9행 6열

```

In [ ]: