

컴피티션 성적분포 및 풀이

여름방학 파이썬스터디 Competition

1) 아래 `string` 이란 변수를 활용하여 문자열 '홀' 만 출력하세요.

(아래 실례에처럼 나와야합니다!// 하나의 문자열로!)

三三三

```
In [1]: string = "홀짝홀짝홀짝"
```

```
In [ ]: # 답안
```

2) 아래 string2 이란 변수를 활용하여 실행 예시와 같이 출력해주세요

실행 예: **터디여름방학파이썬**

```
In [5]: string2 = '여름방학파이썬스터디'
```

In []: # 답안

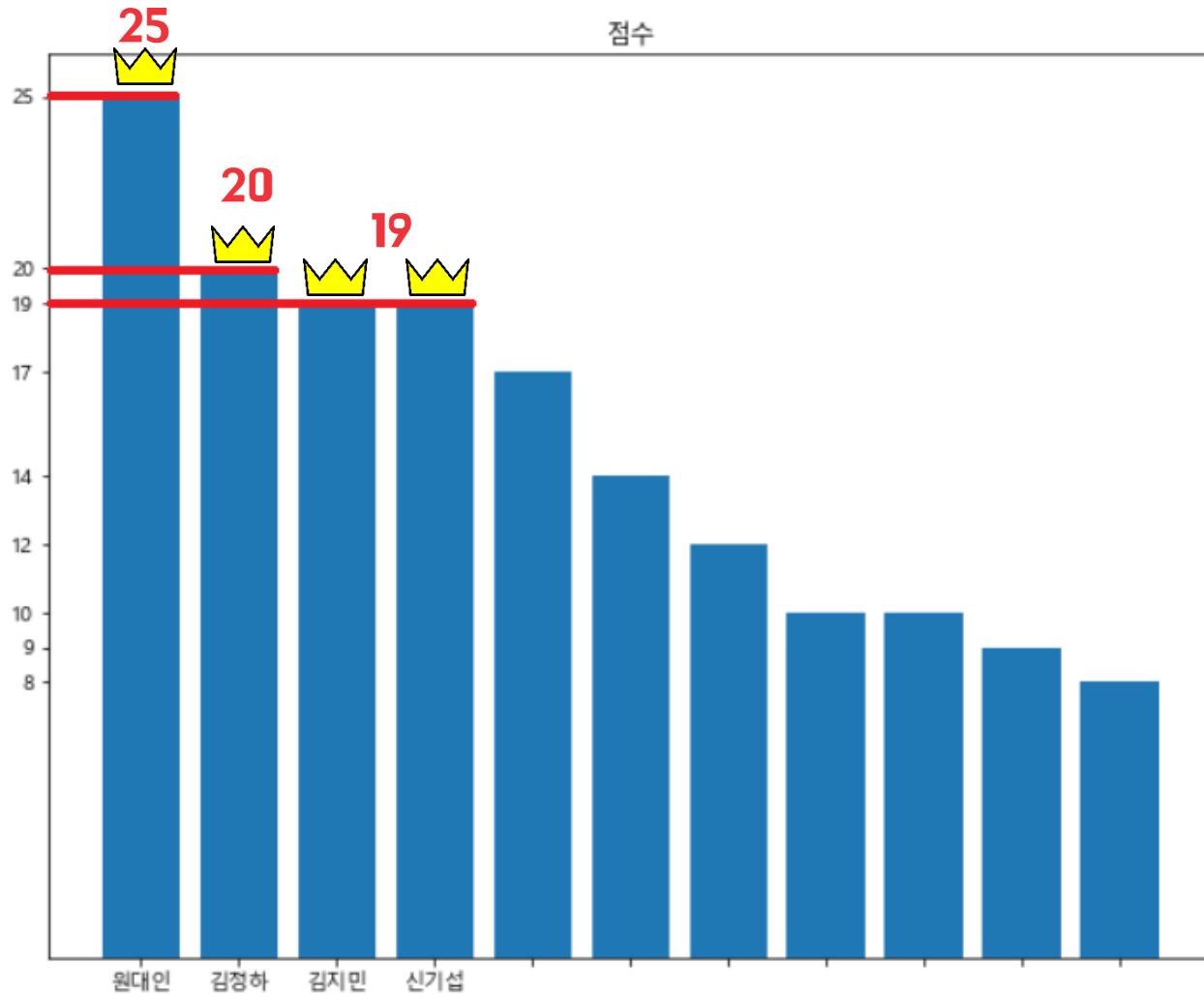
3) 아래 `duplication` 변수를 활용해서 리스트 속 문자열의 중복을 제거해주세요.

```
In [10]: duplication = ['a', 'b', 'c', 'd', 'g', 'h', 'c', 'd', 'l', 'e', 'f', 'g', 'h', 'c', 'd', 'g', 'h', 'c', 'd', 'l', 'c', 'd', 'b', 'c', 'l', 'b', 'c', 'l', 'h', 'c', 'd',
```

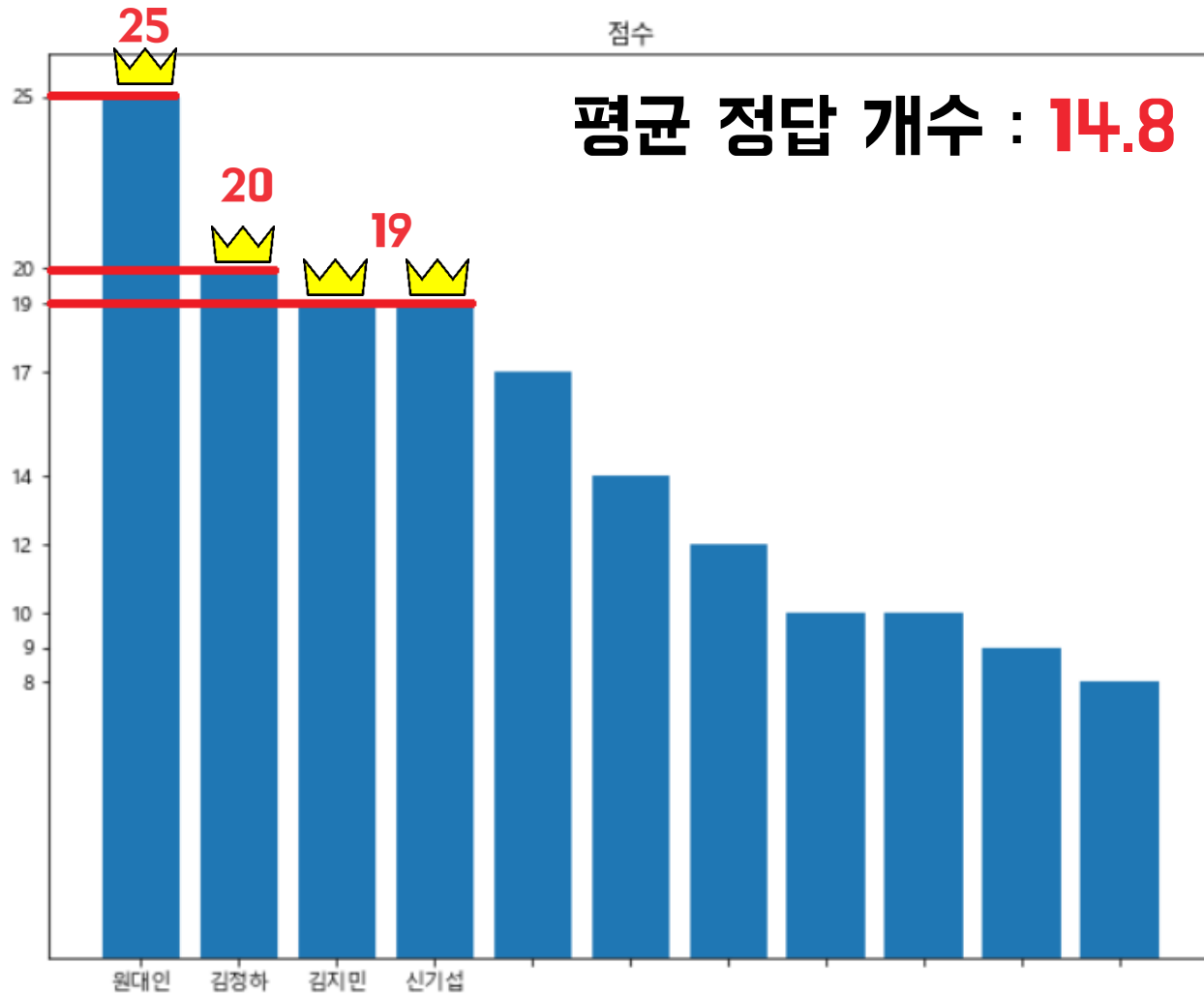
```
In [ ]: # 답안
```

100

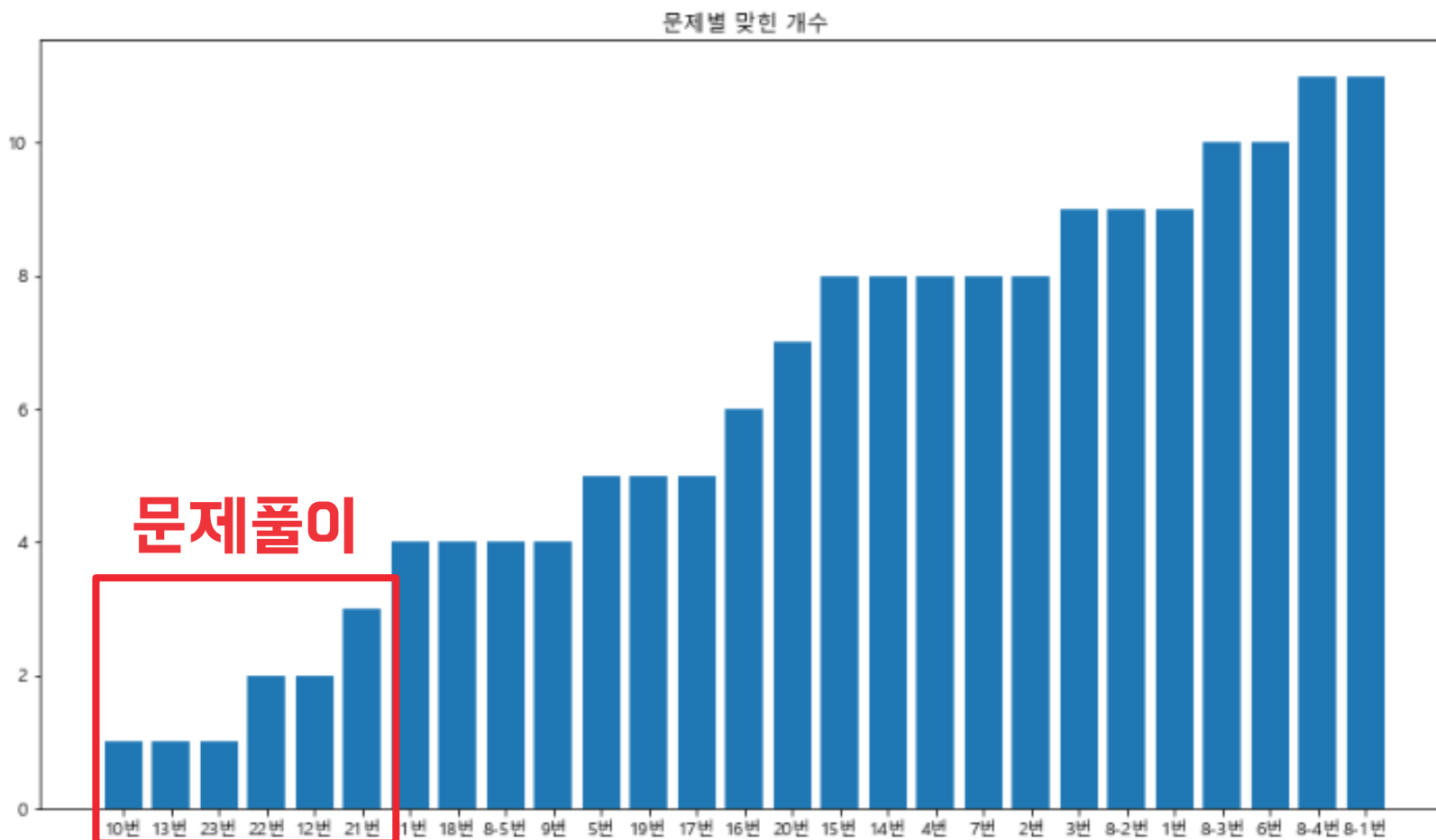
1. 전체성적분포



1. 전체성적분포



2. 문제별 정답자수



3. 문제풀이 - 10번

정답자수 # 1명 # 고난도 # 논리형문제

용돈 11만원에서 매일 3000원씩 사용하되, 7일째 되는 날에는 그 두 배를 사용할 때,
남은 용돈이 있는 가장 늦은 날짜 수를 출력하세요. (while문 사용)

※ 7, 14, 21, ... 일째에 3000원의 두 배인 6000원을 사용하는 것입니다. ※

```
my_money = 110000
days = 0

while my_money > 0:
    if days%7==0 and days>0: Point!
        my_money -= 6000
    else:
        my_money -= 3000
    days += 1

print('용돈을', days, '일까지 사용할 수 있습니다.')
```

용돈을 33 일까지 사용할 수 있습니다.

3. 문제풀이 - 10번 모범답안

정답자수 # 1명 # 최고난도

원래 답안보다 훨씬 간단하게 코드가 짜여졌다.
'`day=day+1`' 코드를 if문보다 위에 넣어 주어
애초에 `day`의 시작을 1로 만들어 두고 조건문을 돌렸기 때문!

```
# 답안
allowance=110000
day=0
while allowance>0:
    day=day+1
    if day%7==0:
        allowance=allowance-6000
    else:
        allowance=allowance-3000

day
```

33

3. 문제풀이 - 11번

정답자수 # 4명 # append # 리스트 # 추가

입력받은 수가 짝수인지 홀수인지 판별해주는 `eov` 라는 이름의 함수를 `def` 를 사용하여 만들고,
그 함수를 호출하여 1부터 100 사이의 홀수 를 리스트 형태로 출력하시오.

→ 힌트 : `append` 함수 이용

```
def eov(n):  
    if n%2 == 0:  
        return "짝수"  
    else:  
        return "홀수"
```

```
whatIsodd = []  
for i in range(1,101):  
    if eov(i)=="홀수":  
        whatIsodd.append(i)  
    else:  
        continue  
print(whatIsodd)
```

[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

3. 문제풀이 - 12번

정답자수 # 2명 # 최고난도 # 11번변형문제 # 구글링최고

주어진 정수가 소수인지 아닌지를 판단하는 `isprimenumber` 이라는 함수를 `def` 를 사용하여 만들고,
그 함수를 호출하여 2부터 100 사이의 소수를 리스트 형태로 출력하시오.

※ 소수 : 1과 자기 자신만으로 나누어 떨어지는 1보다 큰 양의 정수

```
def isprimenumber(n):  
    count=0  
    for i in range(2,n):  
        if n%i==0:  
            count+=1  
            break  
        else:  
            continue  
    if count>0:  
        return 0  
    else:  
        return 1
```

```
whatisprime = []  
for i in range(2,101):  
    if isprimenumber(i)==1:  
        whatisprime.append(i)  
    else:  
        continue  
print(whatisprime)
```

**앞의 11번 문제와 똑같은 방식이지만
함수를 만드는 게 까다로워졌을 뿐!**

3. 문제풀이 - 13번

정답자수 # 1명 # 고난도 # 데이터타입 # 변환

5개의 정수를 input을 통해 입력받고,
짝수는 짝수끼리 누적 합을 구하고, 홀수는 홀수끼리 누적 합을 구하는 프로그램을 작성하시오.

2, 5, 8, 12, 27 을 입력한 후 결과값을 출력하세요

※ 주의 : input 으로 입력받은 자료의 데이터타입이 무엇일지 생각해 보세요!

```
i = 0
even = 0
odd = 0

while i < 5:
    iStr = input(" 키를 입력하세요 : ")
    number = int(iStr)
    if number % 2 == 0:
        even = even + number
    else:
        odd = odd + number
    i = i + 1

print('짝수의 누적 합은', even, '입니다.')
print('홀수의 누적 합은', odd, '입니다.')
```

3. 문제풀이 - 21번

정답자수 # 3명 # value_counts()

금액이 100만원 이상인 데이터를 뽑은 후, corner 별로 각 몇개의 제품이 존재하는지 설명되어있는 데이터프레임을 출력하세요.

```
df.loc[df.amount>=1000000].corner.value_counts().reset_index()
```

	index	corner
0	수입명품	27
1	가전	25
2	디자이너부띠끄	18
3	장신구	15
4	캐릭터캐주얼	13
5	가구	12

- 1) loc 조건문을 이용하여 df 의 amount
열이 1000000 원 이상인 행들만 추출
- 2) corner 열만 추출
- 3) 범주형 데이터의 범주별 개수를 세 주는
함수인 value_counts()를 이용하여
corner별 제품 개수를 세 준다.
- 4) reset_index() 처리로 깔끔하게!

3. 문제풀이 - 22번

정답자수 # 2명 # reset_index() #꼭해주기

수입품 중 가장 많이 수입되는 5개의 브랜드를 출력하세요.

```
df.loc[df['import'] == 1].brand.value_counts().reset_index().iloc[:5,0]
```

```
0      클리닉  
1      샤넬  
2      에스티로더  
3      크리스찬디올화장품  
4      메이크포에버  
Name: index, dtype: object
```

- 1) loc 조건문을 이용 import==1 (수입품) 추출
- 2) Brand 열만 추출 후 value_counts()
- 3) value_counts() 특성상 가장 많은 것부터 내림차순으로 출력된다!
- 4) reset_index() 처리를 해 주어 수입브랜드 들을 인덱스가 아닌 한 열로 만들어줌
- 5) 가장 위에서 5번째까지 브랜드명(0번째) 추출

3. 문제풀이 - 23번

정답자수 # 1명 # 아마도 # 시간부족 #차근차근

**12개월 할부를 가장 많이 하는 고객의 아이디와
그 고객이 구매한 모든 상품 데이터를 출력하세요.**

```
df1 = df.query('installment == installment.max()')  
df1
```

1) 할부가 12개월인 (최대인) 모든 행 추출

```
mc = df1.custid.value_counts().reset_index().iloc[0,0]  
mc
```

**2) df1 데이터에서의 custid 의 범주별 개수를 value_counts()를 이용하여 세고,
할부를 가장 많이 한 사람(가장 첫번째 줄) 의 아이디를 추출**

```
df.loc[df.custid==mc]
```

3) 그 아이디를 가진 고객의 모든 데이터 추출



수고하셨습니다!