

9월 26일 (일)

1. 엘리스 SW 엔지니어 트랙 코딩테스트 준비 - 풀어보며 익히는 알고리즘

첫 번째 문제들

1) 소수의 개수 세기 - 에라토스테네스의 채 구현

소수의 개수 세기

약수는 어떤 수를 나누어떨어지게 하는 수를 의미합니다.

예를 들어 4는 16의 약수입니다. $16 \div 4 = 4$ 이기 때문입니다.

하지만 3은 16의 약수가 아닙니다. $16 \div 3 = 5 \dots 1$ 으로 1이 남기 때문입니다.

소수(prime number)는 약수가 2개인 수를 의미합니다.

예를 들어 2, 3, 5와 같은 수는 약수가 1과 자기 자신밖에 없으므로 소수입니다.

어떤 수 n 이 입력되면, $1 \sim n$ 사이의 소수가 몇 개인지 세어서 출력하는 프로그램을 작성해 봅시다.

입력

어떤 수 n 이 입력됩니다. 해당 수는 10,000 이하의 자연수입니다.

출력

$1 \sim n$ 인 수 중에서 소수인 수의 개수를 출력합니다.

시간제한

- 채점 중에는 제출된 프로그램을 다양한 입력값으로 여러 번 실행하면서 반복적으로 검사합니다.
- 이 문제에서는 제출된 프로그램을 1회 실행하는데 1초가 주어집니다.
- 만약 그 시간을 넘길 경우 오답 처리됩니다. 시간제한을 넘기지 않도록 효율적으로 작성해야겠지요?

입력 예시

10

Copy

출력 예시

4

Copy

- $1 \sim 10$ 중 소수는 2, 3, 5, 7로 총 **4개**가 있습니다.

```

1  # n의 개수를 입력받는다. ~
2  n = int(input())~
3  ~
4  # n의 개수만큼 True로 구성된 배열을 생성한다. (true는 소수라는 뜻)~
5  array = [True for _ in range(n+1)]~
6  ~
7  # 소수의 개수를 출력할 count변수를 생성한다.~
8  count = 0~
9  ~
10 ~
11 # 에라토스테네스의 체 구현 ~
12 for i in range(2, n+1):~
13     if array[i] == True:~
14         j = 2 # ~
15         while i*j <= n:~
16             array[i*j] = False~
17             j += 1~
18         ~
19 ~
20 # 에라토스테네스의 체로 걸러진 소수들을 출력~
21 for i in range(2, n+1):~
22     if array[i] == True:~
23         count += 1~
24     ~
25 print(count)~

```

기출 문제 모음집 Level 1

2. (구현) 도도새의 업무일지

도도새의 업무일지

도도새는 관광객들에게 판매하기 위해 조개껍데기로 액세서리를 만드는 일을 합니다. 매일 액세서리를 만들지만, 매번 같은 양의 액세서리를 제작하지는 않습니다. 그래서 도도새는 자신이 몇 개의 액세서리를 제작하는지 개수를 기록하는 업무일지를 작성합니다.

업무일지는 두 가지로 작성을 하는데 첫 번째 업무일지 A에는 매일매일 만드는 액세서리의 개수는 그대로 기재합니다. 그리고 두 번째 업무일지 B에는 첫날부터 매일 만들어온 액세서리 개수의 평균값을 기재합니다.

예를 들어, 도도새가 5일간 1개, 3개, 2개, 6개, 8개의 액세서리를 제작했다면 업무일지 A에는 1, 3, 2, 6, 8과 같이 기입하면 되고, 업무일지 B에는 $1/1$, $(1+3)/2$, $(1+3+2)/3$, $(1+3+2+6)/4$, $(1+3+2+6+8)/5$, 즉, 1, 2, 2, 3, 4처럼 입력하면 됩니다.

도도새가 업무일지 A를 잃어버렸습니다. 업무일지 B를 보고 업무일지 A를 복구하는 프로그램을 작성하세요.

[입력]

- 첫째 줄에는 업무일지 B의 길이를 나타내는 정수 N 을 입력합니다.

$$(1 \leq N \leq 100)$$

- 둘째 줄에는 업무일지 B_i 를 이루는 N 개의 정수를 입력합니다.

$$(1 \leq B_i \leq 1,000)$$

[출력]

- 업무일지 A를 이루는 N 개의 정수를 출력합니다.

$$(1 \leq A_i \leq 1,000)$$

[입력 예시]

```
5
1 2 2 3 4
```

Copy

[출력 예시]

```
1 3 2 6 8
```

Copy

```
main.py
1 # 자릿수 n을 입력
2 n = int(input())
3 # 각 평균값을 입력
4 arr = list(map(int, input().split()))
5
6 # 결과값 리스트 생성 및 첫번째 원소는 평균값이나 origin 값이 같으므로 미리 넣어준다.
7 result = []
8 result.append(arr[0])
9
10 # origin 값 구하기
11 for i in range(1, n):
12     origin_value = (arr[i]*(i+1)) - sum(result)
13     result.append(origin_value)
14
15 # origin 리스트 join 하기 (join 은 문자열 리스트에서만 사용이 가능하므로 숫자 리스트는 map()함수를 통해 문자열로 변경해줘야한다. )
16 print(" ".join(map(str, result)))
```

3. (구현) 주방장 도도새

주방장 도도새

도도새는 주문이 들어오면 즉석에서 요리를 만드는 요리사입니다. 도도새의 식당에서는 모든 주문은 주문이 들어온 순서대로 처리되며 요리마다 요리에 걸리는 시간이 정해져 있습니다. 그래서 주문이 많이 밀리면 음식을 받지 못하는 손님도 생길 수 있습니다.

예를 들어, 도도새의 근무 시간이 180분이고, 들어온 주문에 걸리는 시간이 주문이 들어온 순으로 각각 45, 30, 55, 20, 80, 20분이라고 합니다.

이 경우, 단 4개의 주문이 처리될 수 있습니다. 처음 4개의 주문이 완료되는 데 걸리는 시간은 150분으로 근무 시간 내에 처리할 수 있지만, 5번째 주문까지 완료하려면 230분이 걸리기에 처리할 수 없습니다.

도도새는 오늘 최대 몇 개의 주문을 처리할 수 있을지 궁금해했습니다. 오늘 들어온 주문이 주어졌을 때, 도도새가 몇 개의 주문을 처리할 수 있는지 계산하는 프로그램을 작성하세요.

[입력]

- 첫 번째 줄에는 들어온 주문의 수를 나타내는 정수 N 과 도도새의 근무 시간(분 단위) T 를 입력합니다.

$$(1 \leq N \leq 50, 1 \leq T \leq 500)$$

- 두 번째 줄은 N 개의 100 이하인 자연수를 입력하며, 각 주문을 처리하는 데 걸리는 시간을 주문 순서대로 입력합니다.

[출력]

- 주문 순서대로 요리했을 때, 근무 시간 내에 처리할 수 있는 주문의 개수를 출력합니다.

[입력 예시]

```
6 180
45 30 55 20 80 20
```

Copy

[출력 예시]

```
4
```

Copy

```
main.py
1 # 주문수, 근로시간, 각 주문시간 입력
2 orderNum, time = map(int, input().split())
3 orders = list(map(int, input().split()))
4
5 # 받을 수 있는 주문수(count), 소요시간(timeSum) 초기화
6 count = 0
7 timeSum = 0
8
9 # 각 주문 더하기
10 for order in orders:
11     if time > timeSum: # 근로시간이 아직 더 남아있다면
12         timeSum += order # 주문 더 받아서 시간 더하기
13         count += 1
14     if time < timeSum: # 주문 받아서 시간 더했는데 근로시간보다 초과할시 (초과안하면 그냥 다음 반복문 실행)
15         timeSum -= order # 다시 시간과 카운트를 빼준다.
16         count -= 1
17         break
18     elif time == timeSum: # 근로시간과 소요시간 같으면 반복문 중단후 count 출력
19         break
20
21
22 print(count)
23
24
```

4. (구현) 생수통

생수통

엘리스 샘물은 물을 담아 판매하기 위해 생수통을 외부 업체로부터 구매하려고 합니다. 구매하는 방법은 다음과 같습니다.

엘리스 샘물은 외부 업체가 제시한 세 종류의 물통과 두 종류의 뚜껑을 적절히 고르고 생수통을 만들어 달라고 요청을 합니다. 그러면 외부 업체가 물통과 뚜껑을 결합하고 재료비(뚜껑 가격+물통 가격)에서 10원을 추가해 엘리스 샘물에 생수통 하나를 판매합니다.

엘리스 생수통의 가격이 최소가 되도록 하고 싶습니다. 각 물통과 뚜껑의 가격을 찾을 때 생수통 한 개 가격의 최솟값을 출력하는 프로그램을 작성하세요.

[입력]

- 첫 번째 줄부터 세 개의 물통의 가격을 한 줄씩, 두 개의 뚜껑 가격 두 줄을 입력합니다.

※물통의 가격은 100원부터 500원까지이고 뚜껑의 가격은 20원부터 80원까지입니다.

[출력]

- 첫 번째 줄에 엘리스 샘물이 구매할 생수통 한 개 가격의 최솟값을 출력합니다.

[입력 예시]

```
300
200
500
20
30
```

Copy

[출력 예시]

```
230
```

Copy


```
main.py ×
1 # 생수통, 뚜껑 리스트 초기화
2 buckets = []
3 lids = []
4
5 # 생수통 입력받기
6 for _ in range(3):
7     bucket = int(input())
8     buckets.append(bucket)
9
10 # 뚜껑 입력받기
11 for _ in range(2):
12     lid = int(input())
13     lids.append(lid)
14
15 # 최소값 출력
16 answer = min(buckets) + min(lids) + 10
17
18
19 print(answer)]
```

5. (구현) 단백질 바

P 식품 회사는 단백질 바를 제조합니다. 단백질 바에 들어가는 재료는 큐브 형태로 3가지가 있으며 각 큐브의 크기는 같지만, 단백질 함량이 다르다고 합니다.

첫 번째 재료는 땅콩 큐브로 1개당 단백질이 10g 함유되어 있습니다.

두 번째 재료는 닭가슴살 큐브로 1개당 40g의 단백질이 함유되어 있습니다.

마지막 세 번째 재료는 순수한 단백질 큐브로 1개당 250g의 단백질이 함유되어 있다고 합니다.

P 식품 회사에서는 고객이 원하는 단백질 함량에 정확하게 맞춰 단백질 바를 제조한다고 합니다. 다만, 포장을 효율적으로 하기 위해 최대한 작은 개수의 큐브를 사용하여 단백질 바를 만든다고 합니다.

만약, 고객이 한 개의 단백질 바에 300g의 단백질 함유를 원한다면 순수 단백질 큐브 1개, 닭가슴살 큐브 1개, 땅콩 큐브 1개로 제조하면 됩니다. 땅콩 큐브만 30개를 사용해도 되지만 단백질 바가 커지면 포장이 비효율적이므로 그렇게 만들지 않습니다.

고객이 한 개에 Ng 짜리 단백질 바를 원한다면, 한 개의 단백질 바에 들어가는 각 큐브 개수의 최솟값을 구하는 프로그램을 작성하세요.

[입력]

- 고객이 원하는 단백질 바 한 개의 단백질 함량을 나타내는 정수 N을 입력합니다.

$$(1 \leq N \leq 10,000)$$

[출력]

- 한 개의 단백질 바에 들어가는 재료 큐브의 개수를 한 줄에 공백으로 구분하여 출력합니다. 출력의 순서는 땅콩 큐브, 닭가슴살 큐브, 순수 단백질 큐브입니다.
- 만약, 만들 수 없는 경우 -1을 출력합니다.

[입력 예시]

300

Copy

[출력 예시]

1 1 1

Copy

```
main.py ×
1 # 단백질 g 입력
2 protein = int(input())
3
4 # 큐브 리스트 초기화
5 cubes = [250, 40, 10]
6
7 # 답안 리스트 초기화
8 answer = []
9
10 # 그리디 알고리즘 실행
11 for cube in cubes:
12     count = protein // cube
13     if count == 0:
14         answer.append(0)
15     else:
16         answer.append(count)
17
18     protein %= cube
19
20 # 전체 answer 리스트의 합이 0이 아니고, protein이 빠짐없이 채워졌다면 결과값 출력
21 if (sum(answer) != 0) and (protein == 0):
22     answer.reverse()
23     print(" ".join(map(str, answer)))
24 else:
25     print(-1) # 나머지가 남아 생성할 수 없으면 -1 출력
26
```

6. (수학) 숫자 나라 특허 전쟁

숫자 나라 특허 전쟁

숫자 나라에는 숫자 3을 상징하는 삼삼 가문과 숫자 5를 상징하는 오오 가문이 있었습니다.

두 가문은 모든 사람이 자신들의 가문을 상징하는 숫자들을 마구잡이로 사용하는 것이 불만이었습니다. 그래서 두 가문은 각각 자신들의 숫자에 특허를 내기로 하였습니다.

삼삼 가문은 모든 3의 배수에 특허를 내고 오오 가문은 모든 5의 배수에 특허를 내었습니다. 이후 사람들은 3의 배수와 5의 배수를 마음대로 쓸 수 없게 되었습니다.

사람들은 마음대로 쓸 수 없는 수의 크기가 얼마나 되는지 알아보기로 하였습니다.

예를 들어, 10보다 작은 자연수라면 3, 5, 6, 9에 특허가 걸려서 사용하지 못하고, 이것을 모두 더하면 그 크기는 23입니다.

그러면 N보다 작은 자연수 중에서 특허가 걸려있는 수를 모두 더한 크기를 구하는 프로그램을 작성하세요.

[입력]

- 첫째 줄에 자연수 N을 입력합니다.

$$(1 \leq N \leq 100,000)$$

[출력]

- N보다 작은 자연수 중에서 특허가 걸려 마음대로 쓸 수 없는 수들의 합을 출력합니다.

[입력 예시]

10

Copy

[출력 예시]

23

Copy

```
main.py
1 # 숫자 입력
2 n = int(input())
3
4 # 정답리스트 초기화
5 answer = []
6
7
8 for i in range(1, n):
9     if i%3 == 0: # 3으로 나눠지면 리스트에 추가
10         answer.append(i)
11     elif i%5 == 0: # 5로 나눠지면 리스트에 추가
12         answer.append(i)
13
14 print(sum(answer)) # 리스트의 모든 원소 합 출력
```

2. 파이썬 TIL

- 리스트(배열) 순서 거꾸로 뒤집기

`reverse()`, `reversed()` 메소드를 이용해서 리스트에 들어있는 요소들의 순서를 거꾸로 뒤집을 수 있다.

reverse()

이 메소드는 아무런 값도 반환하지 않지만, 리스트에 있는 값의 순서를 거꾸로 뒤집습니다.

예제

```
1 list = [0,10,20,40]
2 list.reverse()
3
4 print(list)
5
6 # [40, 20, 10, 0]
```

reversed()

이 메소드는 순서가 거꾸로 뒤집힌 리스트를 반환합니다.

예제

```
1 seqList = [1, 2, 4, 3, 5]
2 print(list(reversed(seqList)))
3
4 # [5, 3, 4, 2, 1]
5
6
7 array=[0, 10, 20, 40]
8 for i in reversed(array):
9     print(i)
10
11 '''
12 40
13 20
14 10
15 0
16 '''
```