

**자료 구조 과제 1**

**과목명 자료구조**

**담당교수 김희철교수님**

**제출일 20210318**

**전공 컴퓨터전자시스템**

**학번 201904458**

**이름 이준용**

* **문제 1번**

2보다 큰 모든 짝수는 두 개의 소수(prime number)의 합으로 나타낼 수 있다는 Goldbach의 추측이 있다. 예를 들어 10 = 3 + 7, 20 = 7 + 13으로 나타낼 수 있다. 2보다 큰 임의의 짝수를 입력하여 이 짝수를 두 소수의 합으로 표현할 수 있는지를 결정하고 만약 표현할 수 있으면 차이가 가장 작은 두 소수의 합으로 나타내시오. 예를 들어 10 = 3 + 7 = 5 + 5이다. 문제에서 요구하는 결과는 5 + 5이다.

**단, n이 소수인지 판별하는 함수를 정의하고 이를 사용하여야 함**

입력 예

10 // 2000이하 정수

출력 예

5 5

* **알고리즘 및 자료구조**

소수인지 판별하는 함수를 정의하기 위해 고대 그리스 수학자 에라토스테네스가 발견한 에라토스테네스의 체를 공부하게 되었습니다.

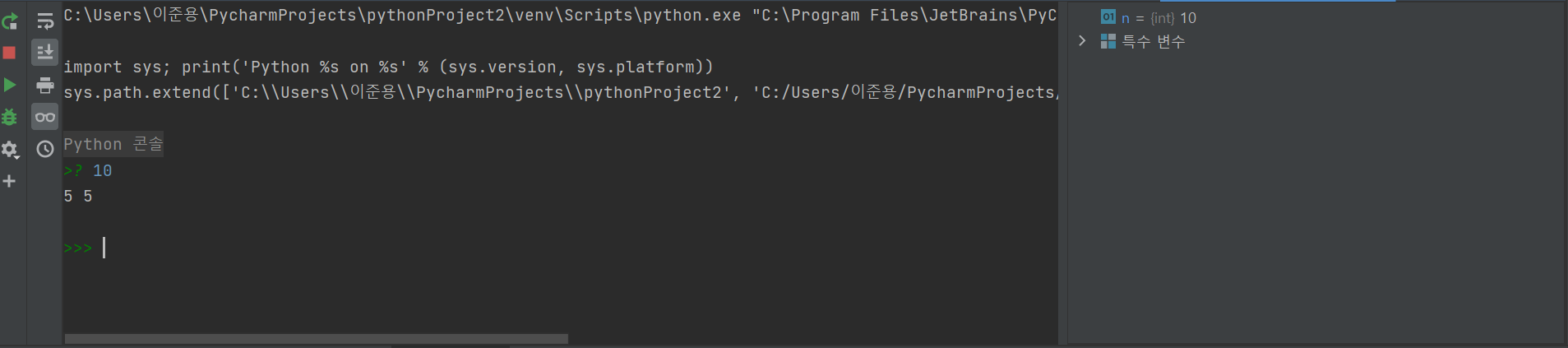
1. 2부터 소수를 구하고자 하는 구간의 모든 수를 나열한다. 그림에서 회색 사각형으로 두른 수들이 여기에 해당한다.
2. 2는 소수이므로 오른쪽에 2를 쓴다. (빨간색)
3. 자기 자신을 제외한 2의 배수를 모두 지운다.
4. 남아있는 수 가운데 3은 소수이므로 오른쪽에 3을 쓴다. (초록색)
5. 자기 자신을 제외한 3의 배수를 모두 지운다.
6. 남아있는 수 가운데 5는 소수이므로 오른쪽에 5를 쓴다. (파란색)
7. 자기 자신을 제외한 5의 배수를 모두 지운다.
8. 남아있는 수 가운데 7은 소수이므로 오른쪽에 7을 쓴다. (노란색)
9. 자기 자신을 제외한 7의 배수를 모두 지운다.
10. 위의 과정을 반복하면 구하는 구간의 모든 소수가 남는다.

<https://ko.wikipedia.org/wiki/%EC%97%90%EB%9D%BC%ED%86%A0%EC%8A%A4%ED%85%8C%EB%84%A4%EC%8A%A4%EC%9D%98_%EC%B2%B4>

* **느낌점**

편입 후 전적대에서는 수강하지 못했던 자료구조수업의 첫 과제였습니다. 파이썬 언어 노베이스 상태여서 0312(금)내내 이문제를 풀기위해 파이썬 문법과 함수들을 찾아보면서 시간을 제일 많이 투자한 문제였습니다. 파이참에서는 실행과 디버그가 문제없이 돌아갔지만 편입 후 처음 사용해보는 goorm으로 과제제출은 더 많이 시간을 투자하게 되었습니다. Runtime error로 인해서 실행과 테스트만 5시간 넘게 한 것 같습니다. 다행히 첫 문제는 통과했지만 다음 문제들은 계속 runtime error로 인해서 파이썬의 runtime error의 종류의 대해서도 깊이 공부하게 되었지만 파이썬이 처음인 저에게는 어디에서 메모리가 낭비되고 있는지 파이썬 문법은 맞게 썼는지 등 모르는게 많았습니다. 처음 받아보는 어려운 과제를 통해서 혼자서 코딩하는 시간의 절대량도 많아지고 파이썬에 대해서 주말 내내 공부하게 되었습니다**.**

* **프로그램 코드**
* # 1.제목:한국외대 자료 구조 과제  
  # 2.날짜:20210312  
  # 문제 1-1  
  # 2보다 큰 모든 짝수는 두 개의 소수(prime number)의 합으로 나타낼 수 있다는 Goldbach의 추측이 있다.  
  # 예를 들어 10 = 3 + 7, 20 = 7 + 13으로 나타낼 수 있다.  
  # 2보다 큰 임의의 짝수를 입력하여 이 짝수를 두 소수의 합으로 표현할 수 있는지를 결정하고  
  # 만약 표현할 수 있으면 차이가 가장 작은 두 소수의 합으로 나타내시오. 예를 들어 10 = 3 + 7 = 5 + 5이다.  
  # 문제에서 요구하는 결과는 5 + 5이다.  
  # 단, n이 소수인지 판별하는 함수를 정의하고 이를 사용하여야 함  
    
  # 입력 예  
  # 10 // 2000이하 정수  
    
  # 출력 예  
  # 5 5  
    
  # n이하의 숫자들 중 소수 찾기  
  def Find\_pnbn(n): # Find prime numbers below n 줄임말 --- 에라토스테네스의 체  
   # 에라토스테네스의 체 초기화: n개의 요소에 True 설정 (소수로 간주)  
   era = [True] \* n  
   # n의 최대 약수가 sqrt(n) 이하이므로 i = sqrt(n)까지 검사  
   m = int(n \*\* 0.5)  
   for i in range(2, m + 1):  
   if era[i] == True: # i가 소수인 경우  
   for j in range(i+i, n, i): # i이후 i의 배수들을 False 판정  
   era[j] = False  
   # 소수 목록 산출  
   return [i for i in range(2, n) if era[i] == True]  
    
  # n 이하의 소수들 중 합이 n  
  # Sum\_opnin 함수 설명: n 이하의 소수들 중 합이 n 이어야 하며 그 차이가 가장 작은 것을 return  
  # 찾은 소수들 중 max(n/2보다 작은 수)부터 합이 n이 되는 수를 찾는다.  
  def Sum\_opnin(n):  
   # Sum of prime numbers is n 줄임말  
   sosu = Find\_pnbn(n) # n 이하의 숫자들 중 소수 찾기 함수에서 소수 받아오기  
   sosulen = max([i for i in range(len(sosu)) if sosu[i] <= n/2]) # 변수 sosulen에 최대 값의 길이를 반환함  
   for i in range(sosulen, -1, -1): # 길이만큼 for구문을 두개를 써서 하나는 순방향  
   for j in range(i, len(sosu)): # 다른 하나는 역방향으로 돌려서  
   if sosu[i]+sosu[j] == n: # n 이하의 소수들 중 합이 n  
   return [sosu[i], sosu[j]] # 합이 n 이면 리턴  
    
  n = int(input())  
  print(" ".join(map(str, Sum\_opnin(n))))



* **문제 2-1번**

2-1) 4지 선다형 문제들의 정답과 문제별 배점이 주어질 때, 제출한 답안지의 점수를 계산하는 프로그램을 작성하시오.

입력 예:

2 4 1 3 3 2 # 문제별 정답 (1번 문제 정답: 2, 2번 문제 정답: 4, ..., 6번 문제 정답 2)

3 3 3 3 4 4 # 문제별 배점 (1번 문제 점수: 3, 2번 문제 점수: 3, ..., 6번 문제 점수 4)

1 4 3 4 1 2 # 제출한 답안지 (1번 문제: 1, 2번 문제: 4, ..., 6번 문제: 2)

출력 예

7 # 정답을 맞춘 문제: 2번, 6번

* **알고리즘 및 자료구조**

제출한 답안지는 1명이므로 학생수(count)는 1로 초기화 했습니다.

클래스 Stu를 생성

inputs함수, points함수, \_\_init\_\_함수, calc함수 총 4개 사용했습니다.

**inputs함수**: 문제별 정답을 입력 받아 리스트로 만듭니다.

**points함수**: 문제별 배점을 입력 받아 리스트로 만듭니다.

**\_\_init\_\_함수**: 학생들의 답안지를 입력 받아 리스트로 만들고 답안지의 개수에 따라 각각의 학생별로 객체 초기화하는 함수 사용했습니다.

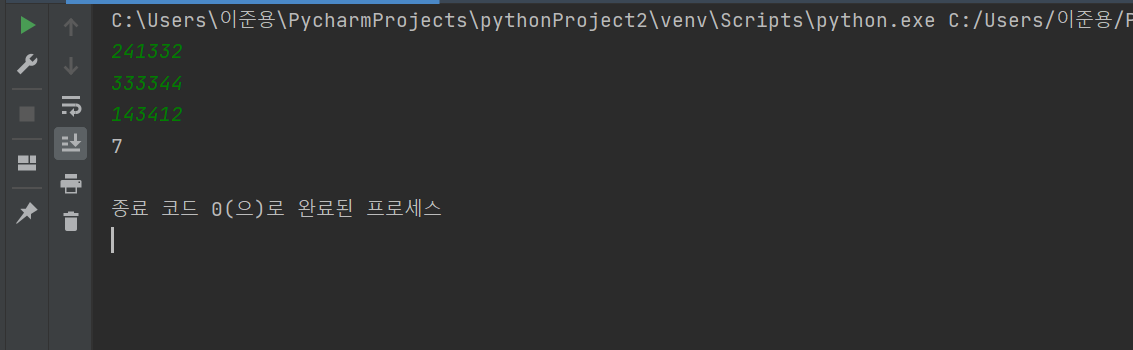
**calc함수**: 정답과 답안지의 답이 같으면 문제의 배점을 더하는 계산 함수입니다.

* **느낀 점**

이 문제부터 runtime error를 잡기위해서 주말 내내 파이썬 문법공부와 한 줄씩 디버깅하는 작업을 계속 했습니다. 하지만 정말 어려웠던 것은 메모리 낭비와 잘못된 참조, 반복된 코드를 고치는 것이 매우 어려웠던 것 같습니다. Goorm에 과제 제출하면서 알게 된 점은 제가 프로그래밍을 하면서 코드 사이사이에 중간점검을 위해 print(변수)를 꼭 넣어서 코드가 잘 돌아가는지 확인용으로 넣었습니다. 근데 계속된 fail로 인해서 무엇이 문제인지 알기 위해서 코드 사이에 삽입된 print() 함수를 모두 지우고 문제에 맞게 입력 값과 출력 값만 그대로 나올 수 있게 만들었습니다. 하지만 2번 문제는 runtime error가 해결되지 않았습니다. 전적대에서는 c언어과 c++, 리눅스 언어로 프로그래밍 할 때는 구조와 스택, 여러 error에 관해서는

다른 IDE환경에서 잡아 주었기에 크게 어려운 점을 못 느꼈습니다. 하지만 2번 문제 덕분에 저의 코드 구조에 대해서 열심히 수정하려고 노력하게 되었습니다.

* **프로그램 코드**
* # 1.제목:한국외대 자료 구조 과제  
  # 2.날짜:20210317  
  # 2-1) 4지 선다형 문제들의 정답과 문제별 배점이 주어질 때, 제출한 답안지의 점수를 계산하는 프로그램을 작성하시오.  
    
  # 입력 예:  
  # 2 4 1 3 3 2 # 문제별 정답 (1번 문제 정답: 2, 2번 문제 정답: 4, ..., 6번 문제 정답 2)  
  # 3 3 3 3 4 4 # 문제별 배점 (1번 문제 점수: 3, 2번 문제 점수: 3, ..., 6번 문제 점수 4)  
  # 1 4 3 4 1 2 # 제출한 답안지 (1번 문제: 1, 2번 문제: 4, ..., 6번 문제: 2)  
    
  # 출력 예  
  # 7 # 정답을 맞춘 문제: 2번, 6번  
  # -\*- coding: utf-8 -\*-  
    
  class Stu:  
   def inputs(self): # 문제 정답 답안 입력(숫자 상관 x)  
   Stu.count = len(self) # 정답의 개수에 따라 count변수에 stirng 길이 만큼 초기화  
   Stu.sinput = list(self) # 문제별 정답을 리스트로 변환하여 sinput변수에 초기화  
   return Stu.count, Stu.sinput # count, sinput변수 리턴  
    
   def points(self): # 문제별 베점 입력(숫자 상관 x)  
   Stu.spoint = list(self) # 문제별 배점 string으로 받아서 list로 변환하여 spoint에 초기화  
   return Stu.spoint # spoint변수 리턴  
    
   def \_\_init\_\_(self, num): # \_\_init\_\_메서드는 학생수를 받으면 제출한 답안지를 받고 정답의 개수를 만족하면 n명의 배열 생성  
   self.num = num # 학생수  
   self.tot = 0 # self.tot는 맞힌 문제 총점, 0으로 초기화  
   self.dap = input() # self.dap변수에 제출한 답안지 받기  
   self.sdap = list(self.dap) # string으로 받은 답안지를 리스트로 변환 sdap에 초기화  
    
   def calc(self): # calc 함수는 정답과 답안지의 값이 같으면 self.tot(총점)변수에 배점을 더하는 함수  
   for j in range(len(Stu.sinput)): # 정답수 만큼의 길이로 for구문 돌려서 Stu.input(답안지)와 self.dap(학생답안지)  
   if self.dap[j] == Stu.sinput[j]: # 인데스의 값이 같으면  
   self.tot = self.tot + int(Stu.spoint[j]) # 해당하는 문제의 배점을 총점에 더해나감  
   print(self.tot)  
  # 정답입력  
  s = input()  
  Stu.inputs(s) # 함수 inputs 호출  
  # 배점입력  
  p = input()  
  Stu.points(p) # 함수 points 호출  
  # 답안입력  
  count = 1 # 2-1 문제에서 정답의 개수와 상관없이 학생수는 1명이므로 1로 초기화  
  li = [] # 제출한 답안지의 학생 수에 따라 li리스트 변수에 학생수 만큼 리스트 생성  
  for z in range(count):  
   li.append(Stu(z + 1))  
  for p in li: # 학생수 만큼 calc 함수 돌려서 총점 확인  
   p.calc()  
  # goorm과제 제출 시스템에서 요구하는 입력값과 출력값을 정확히 처리하기 위해 확인을 위한 중간에 변수  
  # 확인을 위한 print함수는 다 제거 했습니다.  
  # runtime error를 잡지 못했습니다.



* **문제 2-2번**

2-2) 4지 선다형 문항들의 정답과 문항별 배점이 주어질 때, 제출한 n개의 답안지를 채점하여 최저 성적과 최고 성적을 출력하시오.

입력 예

2 4 1 3 3 2 # 문항별 정답

3 3 3 3 4 4 # 문항별 배점

4 # 제출한 답안지 수 n

3 3 3 3 3 3 # 제출한 답안지1

2 2 2 2 2 2 # 제출한 답안지2

4 4 4 4 4 4 # 제출한 답안지3

2 4 1 3 3 2 # 제출한 답안지4

출력 예

3 20

* **알고리즘 및 자료구조**

클래스 Stu를 생성

2-1번과는 다르게 1이상 학생수를 받아야 하므로 학생수 입력 하는 count변수는 메인에서 받았습니다.

inputs함수, points함수, \_\_init\_\_함수, calc함수 총 4개 사용했습니다.

그리고 각 함수마다 while 문을 넣어서 정답의 개수와 입력한 배점, 학생들 답안의 개수가 다를 경우 다시 입력하게 만들었습니다.

**inputs함수**: 문제별 정답을 입력 받아 리스트로 만듭니다.

**points함수**: 문제별 배점을 입력 받아 리스트로 만듭니다.

**\_\_init\_\_함수**: 메인에서 받은 count의 학생들의 답안지를 입력 받아 리스트로 만들고 답안지의 개수에 따라 각각의 학생별로 객체 초기화하는 함수 사용했습니다.

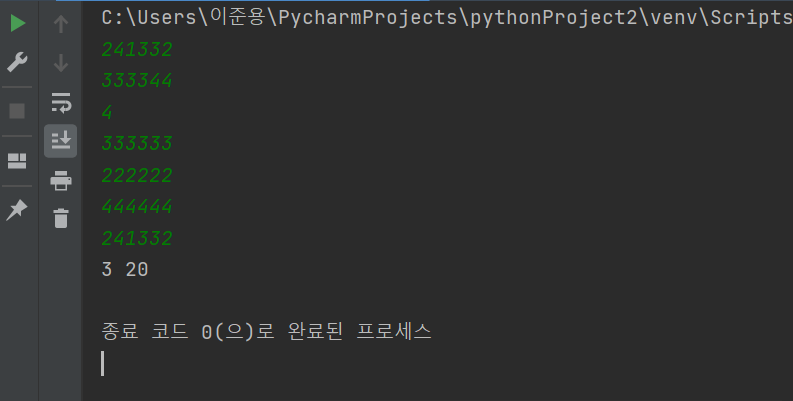
**calc함수**: 정답과 답안지의 답이 같으면 문제의 배점을 더하는 계산 함수입니다.

그리고 calc함수에서 최저점과 최고점을 출력하기 위해 minmaxli 리스트 변수를 만들어 정답과 답안이 맞으면 문제의 배점을 추가하는 구조로 만들었습니다.

* **느낀 점**

2-1번과 마찬가지로 실행해보면서 저의 파이썬 환경에서는 잘 작동했지만 2-1번의 런타임오류가 발생하여 학생들의 답안지에서 문제의 정답을 제출 안하고 빈칸으로 입력을 받았을 수 있겠구나 하여 while문을 고쳐서 입력에 빈칸이 있을 경우 리스트에 빈칸으로 대입하는 코드로 바꿔 봤었지만 여전히 런타임 오류를 고치지 못했습니다.

* **프로그램 코드**
* # 1.제목:한국외대 자료 구조 과제  
  # 2.날짜:20210317  
  # 2-2) 4지 선다형 문항들의 정답과 문항별 배점이 주어질 때, 제출한 n개의 답안지를 채점하여 최저 성적과 최고 성적을 출력하시오  
  # 입력 예  
  # 2 4 1 3 3 2 # 문항별 정답  
  # 3 3 3 3 4 4 # 문항별 배점  
  # 4 # 체출한 답안지 수 n  
  # 3 3 3 3 3 3 # 제출한 답안지1  
  # 2 2 2 2 2 2 # 제출한 답안지2  
  # 4 4 4 4 4 4 # 제출한 답안지3  
  # 2 4 1 3 3 2 # 제출한 답안지4  
  #  
  # 출력 예  
  # 3 20  
    
  class Stu:  
   minmaxli = [] # 점수 최대값, 최소값을 담는 변수  
   def inputs(self): # 문제 정답 답안 입력(숫자 상관 x)  
   Stu.count = len(self) # 정답의 개수에 따라 count변수에 stirng 길이 만큼 초기화  
   Stu.sinput = list(self) # 문제별 정답을 리스트로 변환하여 sinput변수에 초기화  
   return Stu.count, Stu.sinput # count, sinput변수 리턴  
    
   def points(self): # 문제별 베점 입력(숫자 상관 x)  
   Stu.spoint = list(self) # 문제별 배점 string으로 받아서 list로 변환하여 spoint에 초기화  
   while True:  
   if len(Stu.spoint) == Stu.count: # 문제별 배점 수와 정답의 개수가 같으면 함수 실행  
   Stu.totscore = sum(map(int, Stu.spoint)) # 만약 개수가 같을 경우 문제의 총점 계산  
   break  
   return Stu.spoint, Stu.totscore # 문제별 배점과 총점 리턴  
    
   def \_\_init\_\_(self, num): # \_\_init\_\_메서드는 학생수를 받으면 제출한 답안지를 받고 정답의 개수를 만족하면 n명의 배열 생성  
   self.num = num # 학생수  
   self.tot = 0 # self.tot는 맞힌 문제 총점, 0으로 초기화  
   self.dap = input() # self.dap변수에 제출한 답안지 받기  
   while True:  
   if len(self.dap) == Stu.count: # 제출한 학생들의 답안지가 답의 개수와 같을때  
   self.sdap = list(self.dap) # # string으로 받은 답안지를 리스트로 변환 sdap에 초기화  
   break  
    
   def calc(self): # calc 함수는 정답과 답안지의 값이 같으면 self.tot(총점)변수에 배점을 더하는 함수  
   for j in range(len(Stu.sinput)): # 정답수 만큼의 길이로 for구문 돌려서 Stu.input(답안지)와 self.dap(학생답안지)  
   if self.sdap[j] == Stu.sinput[j]: # 인데스의 값이 같으면  
   self.tot = self.tot + int(Stu.spoint[j]) # 해당하는 문제의 배점을 총점에 더해나감  
   Stu.minmaxli += [int(self.tot)] # 정답과 답지가 같을경우 총점 변수에 그 문제의 배점을 더해나감  
  # 정답입력  
  s = input()  
  Stu.inputs(s)  
  # 배점입력  
  p = input()  
  Stu.points(p)  
  # 답안입력  
    
  count = int(input()) # 제출한 답안지의 학생수를 받음  
  li = [] # 제출한 답안지의 학생 수에 따라 li리스트 변수에 학생수 만큼 리스트 생성  
  for y in range(count):  
   li.append(Stu(y + 1))  
    
  for z in li: # 학생수 만큼 calc 함수 돌려서 총점 확인  
   z.calc()  
    
  print(min(Stu.minmaxli), max(Stu.minmaxli)) # 출력값인 최저점과 최고점을 출력  
  # goorm과제 제출 시스템에서 요구하는 입력값과 출력값을 정확히 처리하기 위해 확인을 위한 중간에 변수  
  # 확인을 위한 print함수는 다 제거 했습니다.  
  # runtime error를 잡지 못했습니다.



* **문제 2-3번**

2-3) 4지 선다형 문항들의 정답과 문항별 배점이 주어질 때, 제출한 n개의 답안지를 채점하여 (2) 정답률이 가장 낮은 문제의 번호를 출력하시오. 정답률이 가장 낮은 문제 여러 개일 경우 이 문제들의 번호를 오름차순으로 출력하시오.

입력 예

2 4 1 3 3 2 # 문항별 정답

3 3 3 3 4 4 # 문항별 배점

4 # 체출한 답안지 수 n

3 3 1 3 3 3 # 제출한 답안지1

2 2 2 2 2 2 # 제출한 답안지2

4 4 4 4 4 4 # 제출한 답안지3

1 1 1 1 1 1 # 제출한 답안지4

출력 예

1 2 4 5 6

* **알고리즘 및 자료구조**

클래스 Stu를 생성

inputs함수, points함수, \_\_init\_\_함수, calc함수 총 4개 사용했습니다.

그리고 각 함수마다 while 문을 넣어서 정답의 개수와 입력한 배점, 학생들 답안의 개수가 다를 경우 다시 입력하게 만들었습니다.

**inputs함수**: 문제별 정답을 입력 받아 리스트로 만듭니다. 2-2번과는 다르게 inputs함수에서

- Stu.res\_sublist: 딕셔너리 value 값을 위해 정답의 개수만큼 오름차순 정렬해주기 위해 함수의 보조 리스트 생성

- Stu.totlist: 각각의 문제들의 정답 개수를 갱신하는 리스트 생성

- Stu.ascentotlist: 정답률 오름차순 번호를 출력하기 위해서 만든 리스트 생성

**points함수**: 문제별 배점을 입력 받아 리스트로 만듭니다.

**\_\_init\_\_함수**: 메인에서 받은 count의 학생들의 답안지를 입력 받아 리스트로 만들고 답안지의 개수에 따라 각각의 학생별로 객체 초기화하는 함수 사용했습니다.

**calc함수**: 정답과 답안지의 답이 같으면 문제의 배점을 더하는 계산 함수입니다.

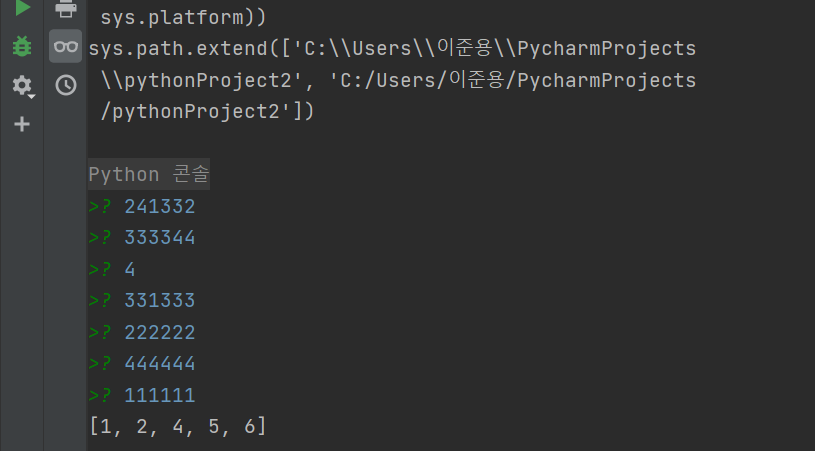
그리고 calc함수에서 ascenli1 = [] 리스트를 생성하여 한 학생의 답안지가 loop구문을 돌때마다 초기화하여 해당 문제 번호를 맞추면 +1되는 코드를 만들었습니다.

**acending함수**: 오름차순 정렬 함수인데요. 딕셔너리를 사용하여 key값은 문제 번호를 주었고, value값은 각 문제의 정답개수를 주었고, 출력은 정답개수를 오름차순으로 정렬하여 key값이 출력되는 함수를 작성하였습니다.

* **느낀 점**

2-2번과 마찬가지로 런타임 오류를 해결하지 못했습니다. 마지막날까지 도전하고 보고서를 작성했으나 한 줄 씩 디버깅해보아도 정답을 찾지 못했습니다.

* **프로그램 코드**
* # 1.제목:한국외대 자료 구조 과제  
  # 2.날짜:20210317  
  # 2-3) 4지 선다형 문항들의 정답과 문항별 배점이 주어질 때, 제출한 n개의 답안지를 채점하여 (2)  
  # 정답률이 가장 낮은 문제의 번호를 출력하시오.  
  # 정답률이 가장 낮은 문제 여러 개일 경우 이 문제들의 번호를 오름차순으로 출력하시오.  
  #  
  # 입력 예  
  # 2 4 1 3 3 2 # 문항별 정답  
  # 3 3 3 3 4 4 # 문항별 배점  
  # 4 # 체출한 답안지 수 n  
  # 3 3 1 3 3 3 # 제출한 답안지1  
  # 2 2 2 2 2 2 # 제출한 답안지2  
  # 4 4 4 4 4 4 # 제출한 답안지3  
  # 1 1 1 1 1 1 # 제출한 답안지4  
  #  
  # 출력 예  
  # 1 2 4 5 6  
  import \_operator  
  import re  
  class Stu:  
   res\_sublist = list()  
   totlist = list()  
   ascentotlist = list()  
   def inputs(self): # 문제 정답 답안 입력(숫자 상관 x)  
   Stu.count = len(self) # 정답의 개수에 따라 count변수에 stirng 길이 만큼 초기화  
   Stu.sinput = list(self) # 문제별 정답을 리스트로 변환하여 sinput변수에 초기화  
   for i in range(1, Stu.count + 1, 1):  
   Stu.res\_sublist.append(1) # 오름차순 정렬해주는 함수의 보조 리스트  
   for t in range(1, Stu.count + 1, 1):  
   Stu.totlist.append(0) # 정답 개수 리스트  
   for x in range(1, Stu.count + 1, 1):  
   Stu.ascentotlist.append(x) # 정답률 오름차순 문제 번호 매기기용  
   return Stu.count, Stu.sinput # count, sinput변수 리턴  
    
   def points(self): # 문제별 베점 입력(숫자 상관 x)  
   Stu.spoint = list(self) # 문제별 배점 string으로 받아서 list로 변환하여 spoint에 초기화  
   while True:  
   if len(Stu.spoint) == Stu.count: # 문제별 배점 수와 정답의 개수가 같으면 함수 실행  
   Stu.totscore = sum(map(int, Stu.spoint)) # 만약 개수가 같을 경우 문제의 총점 계산  
   break  
   return Stu.spoint, Stu.totscore # 문제별 배점과 총점 리턴  
    
   def \_\_init\_\_(self, num): # \_\_init\_\_메서드는 학생수를 받으면 제출한 답안지를 받고 정답의 개수를 만족하면 n명의 배열 생성  
   self.num = num # 학생수  
   self.tot = 0 # self.tot는 맞힌 문제 총점, 0으로 초기화  
   self.dap = input() # self.dap변수에 제출한 답안지 받기  
   while True:  
   if len(self.dap) == Stu.count: # 제출한 학생들의 답안지가 답의 개수와 같을때  
   self.sdap = list(self.dap) # string으로 받은 답안지를 리스트로 변환 sdap에 초기화  
   break  
    
   def calc(self):  
   Stu.ascenli1 = [] # ascenli1리스트를 루프 돌때마다 초기화 시키기  
   for j in range(Stu.count):  
   if Stu.sinput[j] == self.sdap[j] : # 인데스의 값이 같으면  
   Stu.ascenli1 += [int(1)] # 문제 맞으면 +1  
   else:  
   Stu.ascenli1 += [int(0)] # 문제 틀리면 +0  
   # for 구문 다 돌면 문제 개수 만큼의 맞힌 문제개수 리스트 형성  
   Stu.totlist = [x + y for x, y in zip(Stu.totlist, Stu.ascenli1)] # 전역변수리스트와 학생 맞힌 문제 개수 리스트  
   # 같은 자리(index) 값을 합한다.  
    
   def acending(): # 오름차순 Ascending 함수  
   raterank = dict(zip(Stu.ascentotlist, Stu.totlist)) # key 문제번호(Stu.ascentotlist)  
   # value 정답개수  
   res\_list = [i for i, value in enumerate(raterank.values()) if value == min(raterank.values())]  
   # 정답 개수가 최소일때 enumerate함수를 써서 열거하고  
   res\_list = [x+y for x,y in zip(res\_list, Stu.res\_sublist)] # value의 맞게 오름차순 정렬  
   print(str(res\_list))  
    
     
  # 정답입력  
  s = input()  
  Stu.inputs(s)  
  # 배점입력  
  p = input()  
  Stu.points(p)  
  # 답안입력  
    
  count = int(input()) # 제출한 답안지의 학생수를 받음  
  li = [] # 제출한 답안지의 학생 수에 따라 li리스트 변수에 학생수 만큼 리스트 생성  
  for y in range(count):  
   li.append(Stu(y + 1))  
    
  for z in li: # 학생수 만큼 calc 함수 돌려서 총점 확인  
   z.calc()  
    
  Stu.acending() # 오름차순으로 정렬해주는 함수 실행  
    
  # goorm과제 제출 시스템에서 요구하는 입력값과 출력값을 정확히 처리하기 위해 확인을 위한 중간에 변수  
  # 확인을 위한 print함수는 다 제거 했습니다.  
  # 2-1번과 2-2번과 같이 runtime error를 잡지 못했습니다.



* **문제 3-1번**

3-1) 문자열이 회문이면 yes를 출력하고 그렇지 않으면 no를 출력하는 프로그램을 작성하시오. (대/소문자 구분하지 않음)

**단, 문자열이 회문인지 판별하는 함수를 정의하고 이를 사용하여야 함**

입력 예 1

absBa

출력 예 1

yes

입력 예 2

absaba

출력 예 2

no

* **알고리즘 및 자료구조**

Word로 string을 입력 받아 Palindromedis함수를 호출하도록 만들었습니다.

**Palindromedis함수**: string.lower()를 써서 입력 받은 문자열을 모두 소문자로 변경하고

단어의 길이의 반 값을 문자열의 처음과 끝을 검사하는데 인덱스 한 칸 씩 안으로 줄어들면서 검사

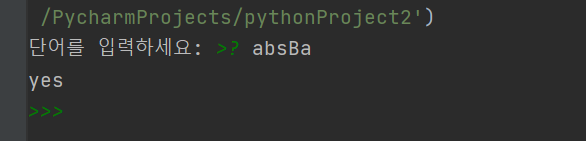
회문이 아니면 no 회문이 맞으면 yes를 출력하는 함수를 작성하였습니다.

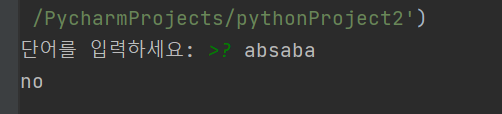
* **느낀 점**

Input()에 아무런 문자도 넣지 않게 되었더니 다행히 제출할 때 pass 하게 되어 다행이었습니다.

특히 회문 판별문제를 이해하고 알고리즘의 논리를 만드는데 시간이 오래 걸렸고 파이썬 기초를 안배워서 쉬운 문제였지만 저에게는 다소 어렵게 느껴졌습니다.

* **프로그램 코드**
* # 1.제목:한국외대 자료 구조 과제  
  # 2.날짜:20210315  
  # 3-1) 문자열이 회문이면 yes를 출력하고 그렇지 않으면 no를 출력하는 프로그램을 작성하시오. (대/소문자 구분하지 않음)  
  #  
  # 단, 문자열이 회문인지 판별하는 함수를 정의하고 이를 사용하여야 함  
  #  
  # 입력 예 1  
  # absBa  
  #  
  # 출력 예 1  
  # yes  
  #  
  # 입력 예 2  
  # absaba  
  #  
  # 출력 예 2  
  # no  
    
  def Palindromedis(string): # 회문 판별 Palindrome discrimination  
   palin = string.lower() # 받은 단어를 소문자로 변경  
   for i in range(0, len(palin)//2): # 단어의 길이의 반 값을  
   if palin[i] != palin[-(i+1)]: # 문자열의 처음과 끝을 검사하는데 인덱스 한칸씩 안으로 줄어들면서 검사  
   return "no" # 회문이 아니면 no 출력  
   return "yes" # 회문이 맞으면 yes 출력  
    
  word = input('단어를 입력하세요: ')  
  print(Palindromedis(word))





* **문제 3-2번**

3-2) 문자열 s의 서로 다른 모든 회문 부문자열(substring)을 사전식 순서대로 출력하는 프로그램을 작성하시오. 단, 대/소문자 구분하지 않으며, 출력하는 회문 부문자열의 각 문자는 소문자이다. sort 함수를 사용하여도 좋음)

**단, 문자열이 회문인지 판별하는 함수를 정의하고 이를 사용하여야 함**

입력 예 1

absAba

출력 예 2

a aba b s

* **알고리즘 및 자료구조**

logestPalindrome함수안에 expend함수를 넣어서 이중으로 함수를 작성하게 되었습니다.

하나는 회문 판별하기 위한 함수이고 다른 하나는 부분문자열이 회문인지 판단하고 맞으면 반환하는 함수입니다.

환문인지 확인하기 위해 인덱스를 체크하며 포인터를 확장하고, 짝수와 홀수 경우 모두 나타나기 때문에 2가지 형태의 포인터 사용했습니다.

* **느낀 점**

파이썬 문법을 공부하다가 함수를 이중으로 작성할 수 있다는 것을 배우게 되었고, 함수를 2중으로 작성하게 되었습니다. 하지만 3-2번은 한글자도 회문이라고 판단하는 함수식을 작성하지는 못했습니다. 출력의 결과는 맞았지만 다른 테스트에서 통과하지 못하였습니다. 보고서 작성할 때까지 시도해보았지만 조금 난이도 있어서 끝내 못 작성하였습니다.

* **프로그램 코드**
* # 3-2) 문자열 s의 서로 다른 모든 회문 부문자열(substring)을 사전식 순서대로 출력하는 프로그램을 작성하시오.  
  # 단, 대/소문자 구분하지 않으며, 출력하는 회문 부문자열의 각 문자는 소문자이다.  
  # sort 함수를 사용하여도 좋음)  
    
  # 단, 문자열이 회문인지 판별하는 함수를 정의하고 이를 사용하여야 함  
    
  # 입력 예 1  
  # absAba  
    
  # 출력 예 2  
  # a aba b s  
    
  from collections import OrderedDict  
    
    
  def logestPalindrome(s: str) -> str: # 회문을 판별하기 위해 함수를 2중으로, 하나는 회문 판별 다른하나는 부분문자열  
   def expand(left: int, right: int) -> str:  
   # 팰린드롬 여부를 체크하며 포인터 확장  
   while left >= 0 and right <= len(s) and s[left] == s[right - 1]:  
   left -= 1  
   right += 1  
   return s[left + 1:right - 1]  
    
   if len(s) < 2 or s == s[::-1]:  
   return s  
    
   result = ''  
    
   for i in range(len(s) - 1):  
   # 팰린드롬은 짝수, 홀수 경우 모두 나타나기 때문에 2가지 형태의 포인터 사용  
   result = max(result,  
   expand(i, i + 1),  
   expand(i, i + 2),  
   key=len)  
    
   return result  
    
  s = input()  
  s = s.lower() # 입력받은 회문 소문자로 변환  
    
    
  s1 = logestPalindrome(s) # 회문 부분문자열 함수  
    
    
  s\_output = (''.join(OrderedDict.fromkeys(s))) # OrderedDict라는 함수를 사용하여 한글자여도 회문인 글자를 추출  
    
  a = list(str(s\_output)) # 회문 인 부분문자열과 list를 합침  
  a.append(s1)  
    
  b = sorted(a) # 오름차순으로 정렬  
    
  print(' '.join(b))  
  # 1번 케이스만 성공  
  # 부분 문자열

