

# 자료구조 4 주차 과제: 파이썬으로 쉽게 풀어쓴 자료구조 4 장

- 이름: 안찬웅
  - 학번: 32162566
  - 과제: 각 문제 당 5 점
1. P4.2
  2. P4.3

\*\* 한 문제라도 컴파일 에러를 해결하지 못하고 제출하는 경우, 전체 과제 0 점. \*\* 풀지 못한 문제 - 만일 과제의 문제를 다 풀지 못한 경우, 여기에 풀지 못한 번호를 적으시오.

과제는 문제에 대한 코딩이 완성되고 테스트를 통해 적절성이 검증된 경우만 점수가 부여되며, 이외 사항에 대해서는 0 점 처리

## 4.2

아래에 코드셀을 만들고, 셀에 함수 testPalindrome 을 작성.

# 특수문자, 공백 제거, 소문자처리

```
def testPalidrome(s):
    stack=[]
    for i in range(len(s)):
        stack.append(s[i])
    for i in range(len(s)):
        if stack.pop() != s[i]:
            return False
    return True
```

# 리스트 생성  
# 문자 s 의 길이만큼 반복  
# s[0]부터 추가  
# pop()이 s[i]가 아니면  
# false 출력  
# 그 외는 True 출력

아래에 코드셀을 만들고, testPalindrome 함수를 호출하여, 입력 "madam, I'm Adam" 과 'race car'를 데이터로 하여 테스트하시오.

```
print(testPalidrome("madam,i'm Adam"))
print(testPalidrome('race car'))
```

```
False
False
```

## 4.3

아래에 코드셀을 만들고, 셀에 4.3 에서 요구된 사항을 반영하여 checkBracketsUpgrade 를 구현하시오. 단 변경사항은 입력은 파일을 읽어 오는 것이 아니고 프로그램의 줄을 요소로 가지는 리스트로 주도록 변경.

테스트: 아래에 코드셀을 만들고, 위에서 작성된 checkBracketsUpgrade 를 다음 입력에 대하여 테스트하시오.

```
myLines = ["{A[(i+1)] = 0}", "if ( (i==0) and (j==0):", "A[(i+1)] =0"]
```

```
class Stack:
    def __init__(self):
        self.top = []
        # 생성자
        # top이 클래스의 멤버 변수가 됨

    def isEmpty(self): return len(self.top) == 0
    def size(self): return len(self.top)
    def clear(self): self.top = []

    def push(self, item):
        self.top.append(item)
        # 스택 s에 item을 삽입

    def pop(self):
        if not self.Empty():
            return self.top.pop(-1)

    def peek(self):
        if not self.isEmpty():
            return self.top[-1]

    def checkBracketsUpgrade(statement):
        stack = Stack()
        for ch in statement:
            # 문자열의 각 문자에 대해
            # in '{[( '도 동일하게 동작함
            if ch in ('{', '[', '('):
                stack.push(ch)
            elif ch in ('}', ']', ')'):
                # in '})]'도 동일하게 동작
                if stack.isEmpty():
                    return False
                    # 조건 2 위반
                else:
                    left = stack.pop()
                    if(ch=="}" and left != "{" or \
                       (ch=="]" and left != "[" or \
                       (ch==")" and left != "("):
                        return False
                        # 조건 3 위반

        return stack.isEmpty()
        # False이면 조건 1 위반

print(checkBracketsUpgrade(["{A[(i+1)] = 0}", "if ( (i==0) and
(j==0):", "A[(i+1)] =0"])))
```

True  
True