실습 7 – 다항식의 덧셈과 곱셈(리스트)

- 실습 목표:
  - 원형 리스트를 이용하여 다항식을 표현하고, 두 다항식의 덧셈과 곱셈 알고리즘을 구현한다.
  - 가용 공간 리스트를 유지하고, getNode, retNode, cerase 등의 알고리즘의 동작 과정을 이해한다.

가용 공간 리스트(Available Space List)

- 프로그램 4.12 ~ 4.14까지의 알고리즘들을 구현
  - avail은 전역 변수로 선언
  - 함수원형 :
    polyPointer getNode(void)
    void retNode(polyPointer node)
    void cerease(polyPointer* ptr)

가용 공간 리스트(Available Space List)

- **(7.1.)** (계수, 지수)의 쌍을 입력받아 원형 연결 리스트로 구현(그림 4.15 참조)하는 함수 A = create_polynomial() 작성
  - 쌍들을 리스트로 구현한 큐에 저장
  - 모든 쌍을 입력받은 후, rear 노드에서 첫번째 노드를 링크
  - 함수원형 :
    polyPointer create_polynomial()
    void attach(float coefficient, int exponent, polyPointer* ptr)

가용 공간 리스트(Available Space List)

- **(7.2.)** main 함수에서는 두 개의 원형 연결 리스트를 생성한 후, 프로그램 4.15의 cpadd를 호출
  - C = cpadd(A, B)
  - 원형 연결 리스트를 출력하는 함수 print_polynomial(C)을 호출하여 실행 결과 확인
  - 함수원형 :
    polyPointer cpadd(polyPointer A, polyPointer B)
    void print_polynomial(polyPointer C)

다항식의 곱셈

- **(7.3.)** 앞에서 구현한 cpadd를 이용하여, 두 다항식을 곱하는 함수 pmul을 작성

- D = cpmul(A, B)의 경우
  - A의 항 하나와 B의 다항식을 곱하는 X = single_mul(A, B)를 구현
  - D = cpadd(D, X)를 이용하여 X를 D에 추가
  - A의 모든 항에 대해 이 과정을 반복
  - 함수원형 :
    polyPointer single_cpmul(polyNode Ai, polyPointer B)
    polyPointer cpmul(polyPointer A, polyPointer B)