Sorted Linked List Code Report – 2017310528 신재광

1. Linked List 관련 함수
   1. InitList – List의 head Node를 malloc을 이용해 메모리를 할당하고, head의 next Node는 빈 상태로 둔다. 또한, List의 len을 0으로 초기화한다.
   2. IsEmpty – List의 len이 0인지 체크하여, 0이면 1, 그렇지 않으면 0을 리턴한다.
2. 16진수 비교
   1. compareHex – Hex\_num 자료형 a와 b를 인자로 받고, 먼저 large를 비교한 후에, large가 같은경우 small을 비교한다. a가 b보다 크거나 같으면 true를 리턴한다.
3. Insert 함수
   1. 변수 pos를 이용해 data가 자신의 위치를 찾아가도록 한다.
   2. 먼저, pos를 0으로 초기화한다.
   3. Node \*자료형 cur, inserting을 선언한다. Inserting은 삽입하는 Node이고, cur는 pos를 측정할 때와 inserting을 삽입할 때 현재 Node의 의미로 사용된다.
   4. Inserting에 데이터를 할당한다.
   5. 만약 Linked List가 비어 있다면(IsEmpty로 확인) head Node의 next Node로 Inserting을 삽입하고, Inserting의 next Node는 NULL로 둔다.
   6. 그렇지 않다면, 데이터들의 대소를 비교해 정확한 위치에 Inserting Node를 삽입해야 한다.
   7. 이를 위해, while문 하나를 선언한다. 종료조건은 cur Node의 next Node가 NULL이거나 (삽입 데이터가 리스트의 모든 데이터 값보다 큼), 데이터가 cur Node의 next Node의 데이터보다 작을 때이다. 즉, 다음 Node의 데이터가 삽입하려는 데이터보다 크거나, 리스트의 끝에 다다를 때까지 반복이 진행된다. 이 반복문을 통해 pos변수로 Inserting이 삽입되어야 할 위치를 지정한다.
   8. cur를 다시 Linked List의 head Node로 초기화한 후, pos 까지 이동시킨다. 그리고, cur 자리에 inserting을 삽입한다.
   9. 두 가지 case로 삽입을 진행한 후에, 리스트의 len 값을 1 올려준다. 구성 Node가 한 개 늘어났기 때문이다.
4. MergeTwoList 함수
   1. cur Node를 선언한다.
   2. cur을 이용해, a list의 모든 데이터를 c list에 삽입한다 (Insert 함수). 이 때 a list는 sorted linked list이므로, 일단은 sorting된 채로 삽입된다.
   3. 이제, 다시 cur을 이용해 b list의 모든 데이터를 c list에 삽입한다. 이 과정 또한 Insert 함수로 진행되기 때문에, a list의 데이터와 비교하여 sorted되면서 삽입된다.
   4. 따라서, 이 두 과정이 끝나면 c list는 자동으로 sorted linked list가 된다.
5. 예시 INPUT : 5 2A EA 1E F0 3B

3 EF 3A 6B

* 1. List a, b에 input값을 Insert한다.
     1. List a에 5개 Node 삽입
     2. List가 비어있으므로, head의 next Node에 데이터 2A 저장
     3. EA 와 2A 비교, EA가 더 크므로 pos ++, cur의 next Node = NULL이므로, pos = 1인 상태에서 반복문 중단
     4. 다음 반복문에 의해 head - 2A – EA로 저장
     5. 1E와 2A 비교 -> 2A가 더 크므로 pos = 0인 상태에서 반복문 중단
     6. 다음 반복문에 의해 head – 1E – 2A – EA로 저장
     7. 같은 일이 반복되어 최종적으로 list a는 head – 1E – 2A – 3B – EA – F0이다.
  2. 같은 일련의 과정에 의해 list b는 head – 3A – 6B - EF이다.
  3. MergeTwoList 함수에 list a, b가 들어간다
     1. 첫 번째 반복문에 의해 list a의 데이터들이 list c에 Insert된다. (list c : head – 1E – 2A – 3B – EA – F0)
     2. 두 번째 반복문에 의해 list b의 데이터들이 list c에 Insert된다. 이 때 Insert의 logic에 의해 자동으로 sorting되어 list c에 들어간다. (list c : head - 1E 2A 3A 3B 6B EA EF F0)
  4. List a, b, c를 line by line으로 출력한다.