**보고서**

**목차**

1. 팀원, 역할 분담 ---------------------------------------------- 3
2. 결과 화면 ---------------------------------------------------- 4
3. 구현 방식 ------------------------------------------------
4. Git 사용 화면 캡쳐

**1. 팀원, 역할 분담**

구한울(조장), 201402316

- item2 구현, 보고서 작성

조우진, 201402434

- item1 구현, 보고서 작성

**2. 결과 화면**

 - car: List의 맨 처음 원소를 리턴 한다.

 - cdr: list의 맨 처음 원소를 제외한 나머지 list를 리턴 한다.

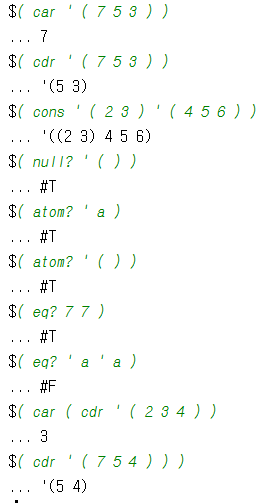
 - cons: 한 개의 원소(head)와 한 개의 리스트(tail)를 붙여서 새로운 리스트를 만들어 리턴 한다.

 - null?: 리스트가 NULL 인지 검사한다. 즉, () 인지 검사한다.

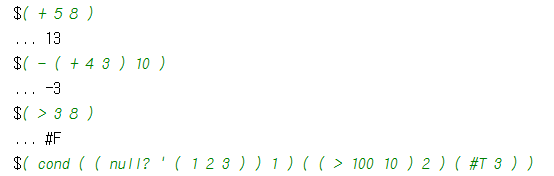
 - atom?: list가 아니면 모두 atom 이다. 따라서 list인 경우는 false, list가 아닌 경우는 true를 리턴 한다.

 - eq?: 숫자 또는 문자열이 같은지 비교하여 true, false를 리턴 한다.

**Item 1. 프로그램의 interpretation 환경 구현**

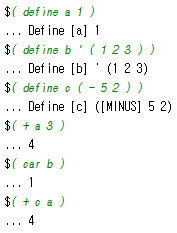




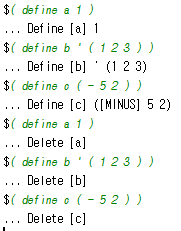


입력 앞에는 “$”표시를 하고, 출력 앞에는 “…” 을 표시하여 구분을 주었다. 테스트 결과 모든 연산이 잘 작동하고 있음을 알 수 있다.

**Item 2. 변수의 바인딩 처리 : define 문으로 변수를 정의하고 사용할 수 있도록 지원한다.**



바인딩 후 연산 사용을 위한 호출 결과



바인딩 후 바인딩 된 변수 바인딩 시도 시 기존 변수의 바인딩 삭제

**3. 구현 방식**

Item 1:

* 프로그램을 계속 실행 시키기 위해 while문으로 무한 루프를 돌리고 exit을 입력해야 종료하도록 했습니다. While 내부 에서는 raw\_input() 함수를 통해 키보드로 입력을 계속 받도록 하였고, 입력 받은 값을 Test\_method(a)를 통해 test코드를 실행시킵니다.

Item 2:

* Python의 기본 자료구조인 Dictionary를 활용하여 구현하였으며 CuteInterpreter 클래스에 심볼테이블을 뜻하는 S\_Table을 선언하고 run\_list에서 define을 인식할 시 수행 될 define\_binding 함수를 선언하여 그 내부에 ID와 value를 심볼테이블에 삽입해줄 insertTable함수를 이미 해당 ID가 S\_Table 에 존재하면 해당 ID의 키 값을 S\_Table에서 지워주고 True를 반환하고 없을 시에는 S\_Table에 ID를 키 값으로 value를 value값으로 하여 저장하고 False를 반환하여 삽입 여부를 define\_binding 함수로 전달하고 define\_binding은 True를 받았을 경우 Type이 ID이고 value를 "Delete [" + rhs1.value + "] "인 Node를 False를 받았을 경우 Type이 define이고 next를 value로 갖는 Node를 반환하여 삽입 여부와 그 키 값, 그리고 삽입 시에는 value까지 모두 출력하도록 구현하였다.

**4. Git 사용 화면**