**20182647 컴퓨터학부 이형규**

**1. 구현 자료구조 및 함수 설명**

**(1) 함수 설명**

void start\_interpreter();

- Interpreter 인터페이스 화면을 실행한다

void file\_load();

- 파일 이름을 입력받아 파일의 내용을 읽어 들인 뒤 수식을 계산하여 결과를 각각 출력한다

void interactive\_mode();

- 한 줄의 수식을 입력 받아 계산 후 결과를 출력한다

char\* ltrim(char\* command);

char\* rtrim(char\* command);

- 입력 받은 문자열의 오른쪽과 왼쪽의 공백들을 제거해준다

void make\_tokens(char\* command, char\*\* tokens, int \*index);

- 입력 받은 문자열을 (, ), MINUS, IF 정수로 각각 토큰화 시켜준다

Node\* make\_tree(char\*\* tokens, int\* index);

- 입력 받은 문자열을 토큰화시킨 토큰 배열을 이용해 트리를 제작하여 root를 리턴한다

Node\* create\_node(char\* name);

- 토큰의 이름을 인자로 받아서 새로운 노드를 생성한다

void print\_tree(Node\* move);

- 구현한 트리를 postorder traversal 방식을 통해 정수일 경우 스택이 저장하고 MINUS 나 IF 일 경우 정수 2개를 꺼내어 연산 결과를 스택에 저장한다

bool is\_character(char a);

bool is\_number(char a);

- 문자가 숫자인지 character인지 판별해준다

bool is\_integer(char\* a);

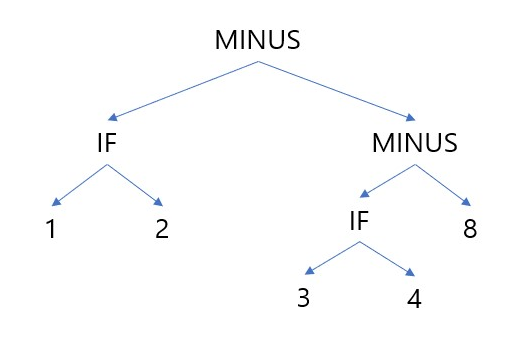
- 인자로 넘어온 문자열이 정수인지 판별한다

bool check\_brackets(char \*\*tokens, int\* index);

- tokens 배열을 확인하면서 ‘(‘ 와 ‘)’ 위치가 올바른지 판별한다

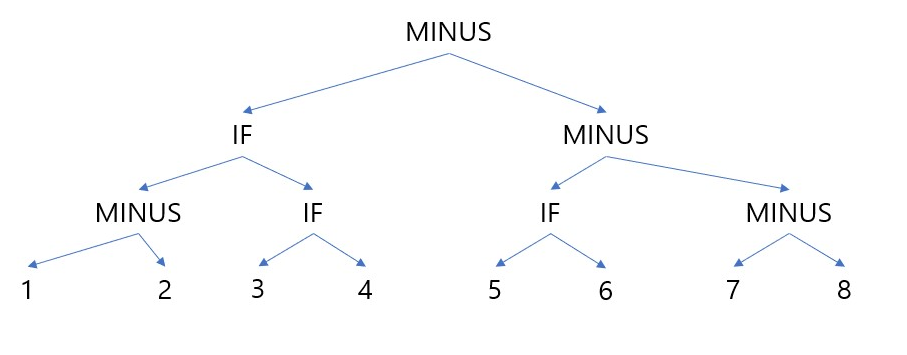
**(2) 구현 자료구조**

**(MINUS (IF 1 2) (MINUS (IF 3 4) 8)의 경우**

****

탐색 순서 : 1->2->IF->3->4->IF->8->MINUS->MINUS 순으로 탐색

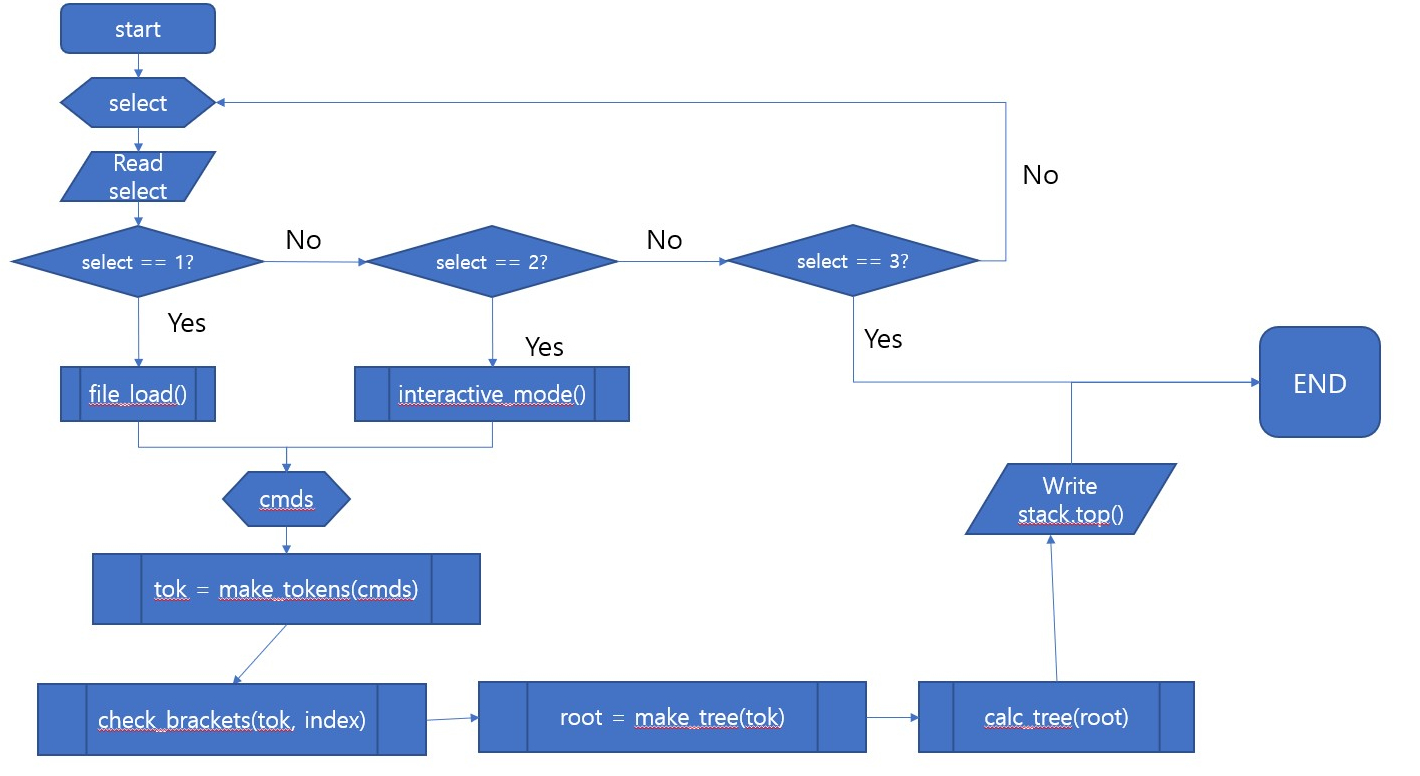
**(MINUS (IF (MINUS 1 2) (IF 3 4)) (MINUS (IF 5 6) (MINUS 7 8)))의 경우**



탐색 순서 : 1->2->MINUS->3->4->IF->IF->5->6->IF->7->8->MINUS->MINUS->MINUS

- 정수일 경우 스택에 저장 IF나MINUS일 경우 스택에서 pop연산을 2번해서 연산 후 다시 스택에 저장. 트리 순회를 마치고 나서 스택에 남아있는 정수 하나가 계산 결과

**2. 흐름도**

****

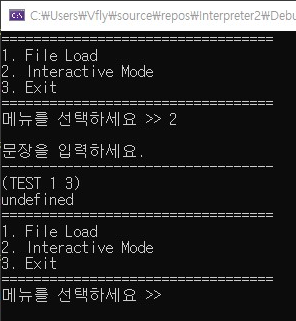
- select가 1번일 경우 file\_load함수 내부에서 cmds입력 -> make\_tokens -> make\_tree -> calc\_tree -> write stack.top()연산이 입력받은 명령어 개수 만큼 반복 됩니다. 자리가 부족해서 반복하는 기호는 생략했습니다.

**3. 실행 결과**

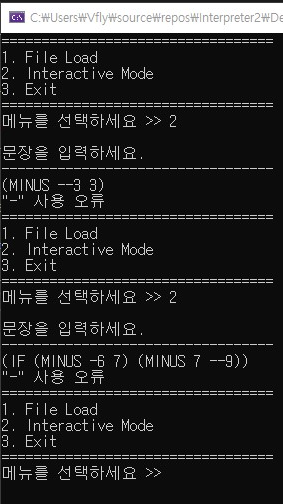
**(Interactive Mode 에서 위의 두 예제 입력 후 실행결과)**

****

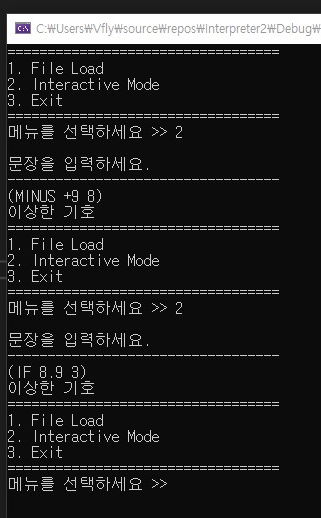
**(undefined 명령어)**

****

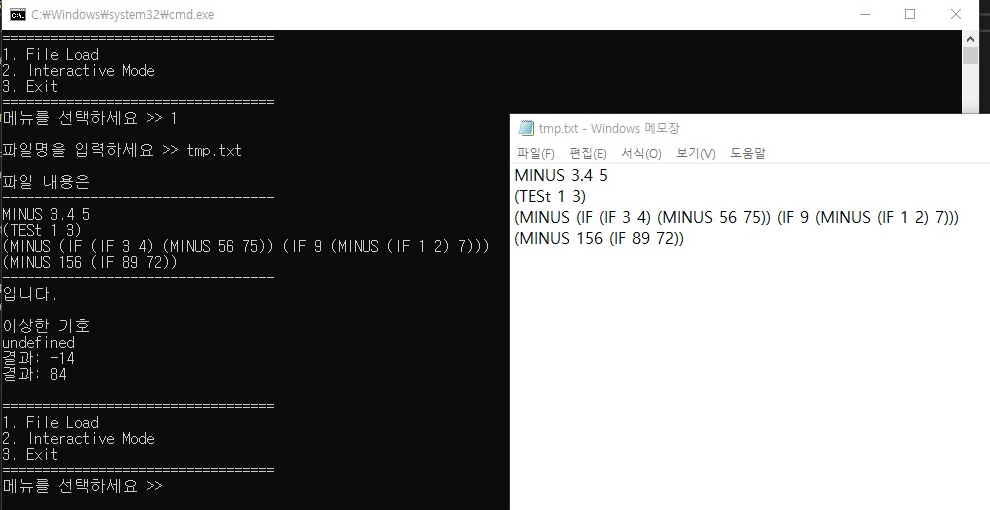
**( ‘ – ‘ 기호 사용 오류 시)**

****

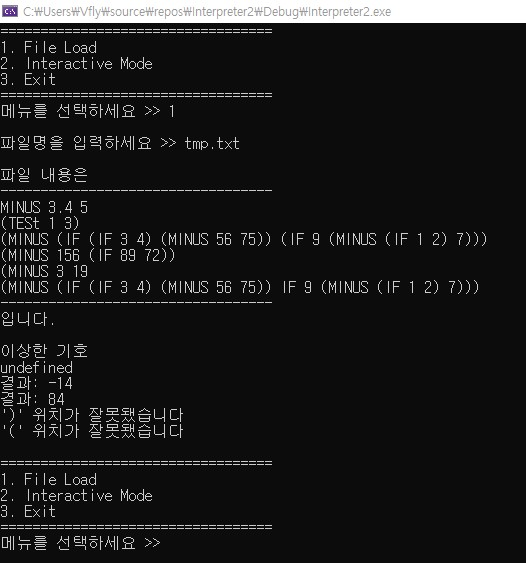
**( ‘ – ‘ 를 제외한 나머지 기호 사용 (실수 사용도 포함되어있습니다))**

****

**(File\_load 시)**

****

**(File load) (괄호 오류 추가)**

****

**- 괄호 오류는 ‘(‘가 있어야될 위치에 없을경우 -> ‘(‘위치 잘못**

**- ‘)’가 있어야 될 위치에 없을 경우 -> ‘)’ 잘못**