

REPORT

Scientific Calculator



교과목 : 프로그래밍2 및 실습

담당교수 : 길아라 교수

학 과 : 컴퓨터학부

학 번 : 20162448

이 름 : 김병준

제출일자 : 2016.10.27(목)

목차

I. 서론

- i. 연구분야
- ii. 동기
- iii. 목적
- iv. 연구접근방법

II. 관련 연구

- i. 자료 탐색
- ii. 우선순위 큐

III. 본론

- i. 프로그램 설명
- ii. 함수

IV. 구현

- i. 실행 화면

V. 결론

I. 서론

i. 연구분야

컴퓨터가 발명된 이래로 인간의 모든 계산은 컴퓨터가 보좌하게 되었다고 말할 수 있다. 그리고 그 이후부터는 점차 모든 문제에 대한 연산을 컴퓨터에게 할당하여 인간의 수고를 덜어주고, 좀 더 정확한 계산을 하기 위한 욕구를 충족시키기 위하여 복잡한 수식을 계산할 수 있는 계산기를 구현하게 되었다. 컴퓨터공학도라면 큐(Queue)와 스택(Stack)을 강의에서 한번쯤 들어보았을 것이다. 큐와 스택은 가장 많이 사용되는 자료구조이다. 큐는 선입선출(FIFO)의 특징을 가지고 있으며, 스택은 선입후출(FILO) 혹은 후입선출(LIFO)의 특징을 가지고 있다. 우리가 주변 어디에서나 접할 수 있는 계산기는 큐와 스택의 대표적인 예시라고 할 수 있다. 계산기는 우리가 평소 사용하는 중위 표기법(Infix notation)이 아닌 후위 표기법(Postfix notation)을 사용하여 주어진 수식을 연산한다. 후위 표기법으로 변환된 수식을 연산하기 위해선 앞에서 말한 큐와 스택을 활용하여 주어진 연산을 수행한다. 우리는 이러한 기본적인 계산기를 넘어서 더 복잡한 수식을 연산할 수 있는 프로그램을 작성하여야 한다.

ii. 동기

큐와 스택으로 계산기를 구현하는 방법은 컴퓨터공학도라면 누구나 알고 있는 사실이다. 그러나 좀 더 복잡한 괄호를 포함한 수식은 단순히 큐와 스택을 사용하는 방법을 알고 있더라도 복잡한 문제라는 것을 우리는 알 수 있다. 또한 우리가 문제를 푸는데 있어서 가장 중요한 것은 변수이다. 중학교 수준 이상의 문제를 접했을 때 $x = 1$ 과 같은 변수 선언을 누구나 한번쯤은 해봤을 것이다. 또한 집합의 경우 $A = \{1, 2, 3, 4, 5\}$ 와 같이 집합의 정의도 하였을 것이다. 그러나 여태까지 제작해본 계산기 프로그램은 보편적으로 쓰이는 $+$, $-$, $*$, $/$ 의 연산자만 지원했다. 또한 변수 및 배열 선언, 활용 또한 불가능했다. 또한 단순히 변환 명세한, 두개를 사용함으로써 여러 개의 연산, 피 연산자를 한번에 처리할 수 없는 문제도 존재하였다. 더 중요한 문제는 변수들의 선언이 계산이 완료되고, 프로그램을 종료한 뒤, 다시 프로그램을 실행하였을 때, 이전에 선언한 변수들을 하나씩 재선언해야한다. 이러한 문제점들은 사용자에게 불편함을 제공하며, 편의성을 해치는 부분이기도 하다.

iii. 목적

본 연구에서는 큐와 스택의 사용과 후위 표기법, 단일 변수 및 배열 변수 선언, 다양한 수학 함수를 지원하여 간단한 공학용 계산기를 구현할 것이다. 그리고 더 복잡한 수식을 연산할 수 있도록 괄호 기능도 구현할 것이다. 또한 저장 및 불러오기 기능을 제공하여 사용자의 변수 선언을 반복하지 않도록 편의성을 제공할 것이다. 마지막으로 TUI 환경에서의 간단한 인터페이스를 제공하며, 메뉴에 도움말 항목을 추가하여 사용자가 계산기를 적극적으로 활용할 수 있도록 도움말을 제공할 것이다.

iv. 연구방법

본 연구에는 C 프로그래밍을 통해 큐와 스택을 구현하고 기본적인 사칙연산($+$, $-$, x , $/$) 구현과 C 시스템이 제공하는 math.h에 포함되어 있는 수학 함수(pow, cos, sin, tan, sqrt, acos, asin, atan, atan2, cosh, sinh, exp, log, log10, ceil, floor, fabs, fmod, ldexp)를 활용 및 구현, 단일 변수 선언 및 사용(A ~ Z), 배열 변수 선언(AA ~ ZZ) 및 사용(ADD, MUL, AVG, STD 함수같은 배열 간의 연산 함수 제공), 파일 입출력을 통해 변수의 저장(save) 및 불러오기(load) 기능을 구현한다. 또한 각 기능들은 함수로 구현되며, 단순 연산과 복합(수식) 연산으로 계산기를 구분한다.

II. 관련 연구

i. 자료 탐색

이산수학에서 배우는 Tree 와 Graph 의 자료구조는 큐와 스택에 대한 이해가 선행되어 있어야 이해가 쉽고, 구현도 할 수 있다. 즉, 큐와 스택을 사용한다고 마음먹고 사용하는 것이 아니라 순서를 고려해야 하는 상황이거나 우선적으로 선택해야 하는 상황에서 많이 사용한다.

ii. 우선순위 큐

우선순위 큐는 조금 특별한 문제를 풀기 위한 것으로 선입선출 개념에서 약간 변형을 한 것이다. 나중에 들어왔지만, 먼저 처리해야 하는 자료가 발생할 경우에 필요로 한다. 이러한 판단은 어떤 상황을 회피하거나, 우선 순위를 두어야 하는 상황에서 사용한다. 그 중에는 대표적으로 환형 큐와 덱(Deque)이 존재한다. 환형 큐는 배열이라는 특성 혹은 제한된 메모리 사용에 맞게 고안된 것이다. 이것은 구현 및 성능에 주의를 기울여야 한다. 덱은 조금 특이한 경우인데, 큐와 스택을 합친 것으로 생각하면 된다. 양쪽 끝에서 삽입 및 삭제가 가능한 자료 구조로 선입선출과 후입선출의 복합적인 성격이 필요한 경우에 사용한다.

III. 본론

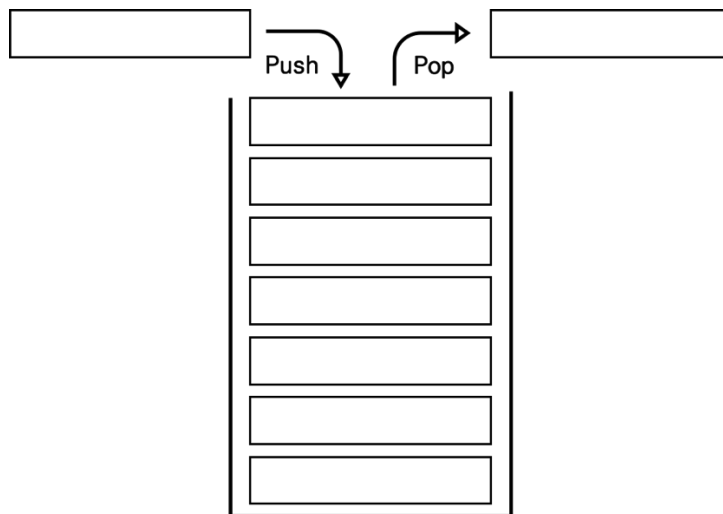
i. 프로그램 설명

우리가 아는 계산기는 $1 + 2 + 3$ 과 같은 형식으로 수식을 입력한다. 우리가 구현하는 계산기 프로그램도 동일한 형식으로 수식을 입력하는데, 이러한 수식 표기를 중위 표기법이라고 한다. 그러나 실제 계산기 내부에서 수식을 연산할 때에는 후위 표기법으로 처리한다. 후위 표기법은 연산자를 연산 대상의 뒤에 쓰는 연산표기법으로 다음과 같이 표시된다.

$$(3 + 5) \times 9 \longrightarrow 3\ 5 + 9\ x$$

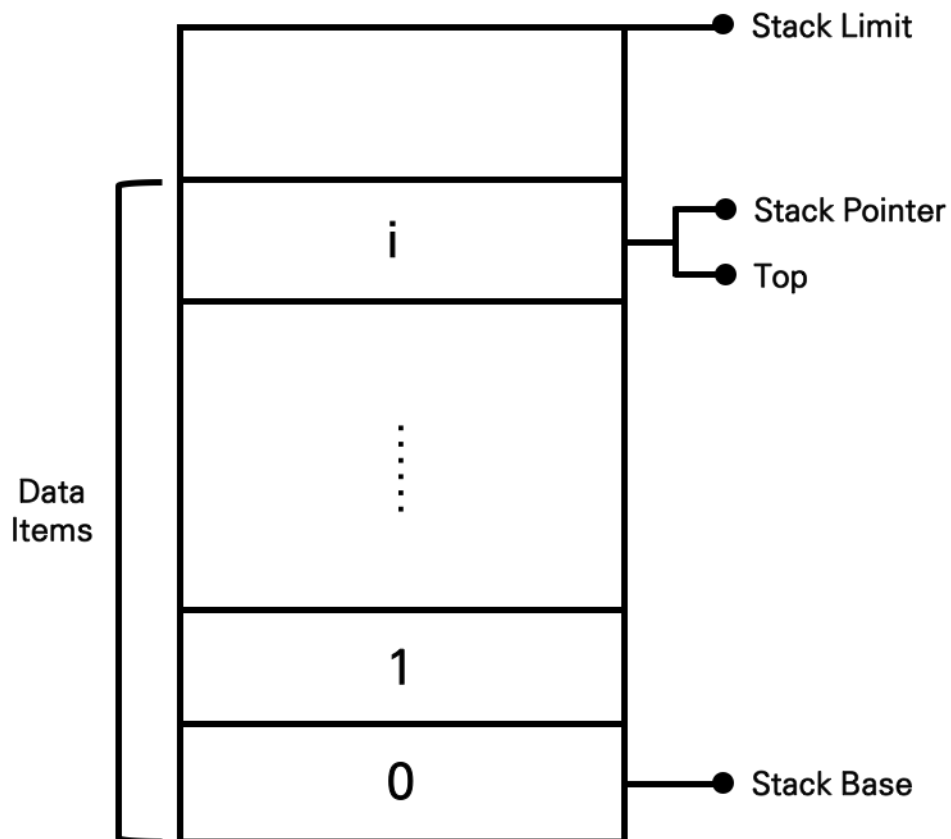
〈그림 1〉 후위표기식 변환

컴퓨터에 존재하는 계산기 프로그램이나 스마트폰에 존재하는 계산기, 물리적 버튼으로 구성된 계산기 등등 모두 이와 같은 형식으로 수식을 처리한다. 프로그래밍을 통해 이를 구현하기 위해선 스택을 활용해야 한다. 우리는 배열로 구현한 스택 자료구조를 사용하여 저장한다. 우선 숫자 부분은 그대로 출력하고 연산자들은 우선순위를 비교하여 출력과 저장여부를 결정하고, 모든 연산자의 삽입(Push)이 끝나면 스택의 모든 값들을 추출(Pop)한다. 그 후 수식의 숫자를 연산자가 나올 때까지 스택에 삽입하고 연산자가 나오면 연산자에 맞는 개수의 숫자를 추출하여 연산하고 그 결과를 다시 스택에 저장한다. 이 과정을 반복하여 수식을 계산한다.



〈그림 2〉 스택의 LIFO(Last In First Out) 방식

스택은 밑에서 쌓아지는 형태로 자료를 저장하며, 삽입과 삭제는 제일 윗부분에서만 일어난다. 꼭대기(Top) 위치를 포인터로 가리키고, 삽입, 추출을 하며 꼭대기를 증가, 감소시킨다.



〈그림 3〉 스택 구조

그런데 우리는 앞으로 변수들을 입력 받아야 한다. 단순히 숫자만 입력 받는 것이 아니라 알파벳을 'A' ~ 'Z'까지 인식하고 선언된 값을 할당해야 한다. 마치 우리가 프로그래밍을 할

때 변수 선언하는 것과 마찬가지로 말이다. 그렇다면 이러한 변수들은 어떻게 인지해야 하고, 어떻게 메모리에 저장해야 하는가를 고민할 필요가 있다. 알파벳 'A'는 ASCII 코드 값으로 65 이다. 알파벳 'B'는 ASCII 코드 값으로 66 이다. 알파벳 또한 연속된 숫자를 가지고 있다. 우리는 연속된 메모리를 가질 수 있는 배열을 이용하여 이 문제를 해결할 수 있다.

A 의 ASCII 코드 값을 기준으로 하여, 'D'라는 변수가 주어졌을 경우, 'D'를 변환 명세를 통해 ASCII 코드 값으로 변환하고 기준 값인 A 의 ASCII 코드 값인 65 를 뺄셈 해주면 배열의 인덱스 값을 얻을 수 있다. 이 인덱스 값을 통해 알파벳에 따라 값을 할당할 수 있으며, 또한 이와 같은 방식으로 계산기 프로그램에서 배열 변수 또한 구현한다.

ii. 함수

- `int main(void);`

`print_menu` 함수를 호출하여 인터페이스를 호출하고, `do-while` 문을 반복하며 메뉴 선택을 입력 받는다. 입력된 값에 따라 해당 함수를 호출한다. '1' 입력 시 `simple_cal` 함수를 호출하고, '2' 입력 시 `formular_cal` 함수 호출, '3' 입력 시 `help` 함수를 호출, '0' 입력 시 프로그램을 종료한다.

- `void simple_calculator(void);`

`print_menu` 함수를 호출하여 인터페이스를 호출하고, 수식을 입력 받아 처리한다.

"!@# \$" 입력 시 함수를 종료하고 홈으로 돌아간다, 다른 경우에는 `simple_to_postfix` 함수로 인자 전달 한 뒤, 반환 값을 비교하고, `Calculation` 함수에 후위표기법의 수식을 인자로 전달한 뒤, 반환 값을 결과 값으로 출력한다.

- `void complex_calculator(void);`

`print_menu` 함수를 호출하여 인터페이스를 호출하고, `while` 문을 돌며 수식을 입력 받아 처리한다. "!@# \$" 입력 시 함수를 종료하고 홈으로 돌아간다. "save 파일명"을 입력할 경우 `save_file` 함수 호출하고 "load 파일명"을 입력할 경우 `load_file` 함수를 호출한다. 이 외의 경우에는 `preprocess` 함수를 호출하여 수식을 처리한다. 종료 시에는 단일, 배열 변수의 메모리를 해제하는 `m_dealloc` 함수를 호출한다.

- `int complex_postfix(char position, char *destination);`

입력 받은 수식을 중위표기법에서 후위표기법으로 변환하는 함수이다. 먼저 단일 변수와 배열 변수일 경우 선언 여부를 검사하고, 문자열의 변수 부분을 ASCII 코드 값으로 변환시켜준다. 수학 함수와 배열 변수 함수일 경우에도 먼저 입력된 함수 수식 부분을 검사하고, 문자열의 해당 부분을 ASCII 코드 값으로 바꾸어 준다. 그 후에는 괄호를 검사해 괄호의 위치를 확인 해 오류를 확인한다. 다음으로 연산자들을 검사해 우선 순위에 따라 삽입해주고, 숫자 또한 삽입한다. 한번 더 괄호의 개수와 연산자와 피 연산자의 개수를 검사하여 오류가 없으면 스택의 내용을 모두 추출해서 후위 표기법 수식을 완성한다.

- `int simple_postfix(char position, char * destination);`

단순 연산에서.입력 받은 수식의 예외 처리(변수 선언 및 괄호 선언 등)와 중위표기법에서 후위표기법으로 변환하는 함수이다.

- `void preprocessor(char *input);`

입력 받은 수식에 단일 변수, 배열 변수가 사용되는 경우 그 변수들을 먼저 처리해 주기 위한 함수이다.

- `int operator_precedence(int operator);`

연산자의 우선 순위를 `int` 형 값으로 반환하는 함수이다.

- `void init_stack(void);`

스택을 초기화하는 함수이다.

- `double pop(void);`
스택의 요소를 추출하는 함수이다.
- `double push(double position);`
스택의 요소를 삽입하는 함수이다.
- `double top(void);`
스택의 최상단의 값만 반환하는 함수이다.
- `void math(char *function, char *argument);`
수학 함수와 배열 변수 함수 처리 함수. POW, COS, SIN, TAN, SQRT, ACIS, EXP, LOG, CEIL, FLOOR, FABS, FMOD 함수를 제공하며 각 함수가 필요한 인자의 개수가 1 개인지 2 개인지 확인하여 각 인자를 문자열 형태로 `to_postfix` 함수에 넣어 변수 혹은 연산처리를 거치고 최종적으로 `calculation` 함수에 전달해 수학 함수의 값을 반환한다. 배열 변수 함수 연산 시 모든 값의 합, 곱, 평균, 표준편차를 반환한다.
- `double calculation(char * position);`
`preprocess` 함수와 `to_postfix(simple_to_postfix)` 함수에서 전 처리를 한 수식을 계산하는 함수이다. 실수 연산을 위해 먼저 숫자 부분을 정수부와 소수부로 나누어 처리하고, 빼기 연산과 음수 기호 '-'를 구분해주고, 나머지 연산자들을 처리해 계산하여 연산 결과를 반환한다.
- `void assignment_variable(int index, char * expression);`
배열 변수의 대입 연산 함수이다.
- `void function_argument(char *position, char input_argument
[][LARGESIZE], int count);`
수학 함수의 인자를 추출하여 문자열 형태로 저장하는 함수이다.
- `void array_calculation(char *expression);`
배열 변수의 오류 처리와 사칙 연산 함수이다.
- `void deallocation (void);`
대입 연산과 파일 불러오기를 통해 동적 할당한 단일, 배열 변수의 메모리 공간을 해제하는 함수이다.
- `void save(char * position);`
단일, 배열 변수가 값을 인자로 받은 문자열의 파일명을 이름으로 하는 파일로 저장한다. 단일 변수가 존재 시 파일에 변수 값을 작성 및 개행하고 존재하지 않을 시 NULL 값을 작성 및 개행한다. 배열 변수가 존재할 시에는 해당 배열 변수에 저장되는 총 값의 개수를 먼저 파일에 작성한다. 그 뒤, 각 값들을 하나씩 쓰며 개행하며, 배열 변수가 존재하지 않을 시 NULL 값을 저장한다.

- `void load(char * position);`
인자로 받은 문자열의 파일명을 이름으로 하는 파일에 저장된 단일, 배열 변수 값을 불러온다.
- `void print_menu(int select_menu);`
인터페이스를 출력해주는 함수이다.
- `void help(void);`
도움말의 메인 함수이며 도움말을 출력한다.
- `void information(void);`
프로그램 제작자의 정보를 출력한다.
- `void terminal_clear(void);`
터미널을 지워준다.
- `void print_error(int type);`
오류를 출력한다.
- `void remove_enter(void);`
엔터 버퍼를 지워준다.

IV. 구현

i. 실행 화면

- 메인 화면

```
● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

Scientific Calculator - Home

== Main Menu =====
[1] 단 순 연 산
[2] 수 식 연 산
[3] 도 움 말
[0] 종 료

=====

입 력  : 
```

〈그림 4〉 메인 화면

- 기본적인 사칙 연산

```
● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

Simple Scientific Calculator - Main

입 력  :  1.1 + 2 * 3
= 7.1
아 무 키 나 입 력 하 세 요 ...

```

〈그림 5〉 기본적인 사칙 연산

- C 시스템이 제공하는 수학 함수 제공

```

● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

Formular Scientific Calculator - Main

입력 : POW(3,4)
= 81
입력 : SQRT(10000)+99
= 199
입력 : FABS(-1234)
= 1234
입력 : B = 200
입력 : LOG(B)
= 5.29832
입력 : █

```

〈그림 6〉 C 시스템이 제공하는 수학 함수 제공

- 단일 변수 선언 및 사용

```

● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

Formular Scientific Calculator - Main

입력 : A = 2
입력 : Z = 8
입력 : A + Z
= 10
입력 : POW(A,2)
= 4
입력 : B
ERROR : 변수 'B'가 선언되지 않음
아무 키나 입력하세요 ...
█

```

〈그림 7〉 단일 변수 선언 및 사용

- 배열 변수 선언 및 사용

```
● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

Formular Scientific Calculator - Main

입력 : AA = 1, 2, 3, 4, 5
입력 : ADD(AA)
= 15
입력 : MUL(AA)
= 120
입력 : AVG(AA)
= 3
입력 : STD(AA)
= 1.41421
입력 : ZZ = 10, 20, 30, 40, 50
입력 : AA + ZZ
= 11, 22, 33, 44, 55
입력 : AA - ZZ
= -9, -18, -27, -36, -45
입력 : AA * ZZ
= 10, 40, 90, 160, 250
입력 : AA / ZZ
= 0.1, 0.1, 0.1, 0.1, 0.1
입력 : █
```

<그림 8> 배열 변수 선언 및 사용

- end(END) 입력 시 홈 화면으로 이동

```

● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

Formular Scientific Calculator - Main

입력  : END

● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

Scientific Calculator - Home

== Main Menu =====

[1] 단순 연산
[2] 수식 연산
[3] 도움말
[0] 종료

=====

입력  : █

```

〈그림 9〉 end(END) 입력 시 홈 화면으로 이동

- 단순 변수와 배열 변수 파일 저장 및 불러오기 기능

```

● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

Formular Scientific Calculator - Main

입력 : A = 1
입력 : B = 2
입력 : C = 3.3
입력 : D = 444
입력 : E = 5550.55
입력 : AA = 1, 2, 3, 4, 5
입력 : BB = 10, 20, 30, 40, 50
입력 : CC = 1.1, 2.2, 3.3, 4.4, 5.5
입력 : save good
입력 : ^C
x ~/Desktop/프로그래밍2 및 실습/ScientificCal

● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

Scientific Calculator - Home

== Main Menu =====
[1] 단순 연산
[2] 수식 연산
[3] 도움말
[0] 종료
=====

입력 : 2

● University : Soongsil University
● Major      : Computer Science & Engineering
● Student ID : 20162448
● Name       : Kim Byoung June

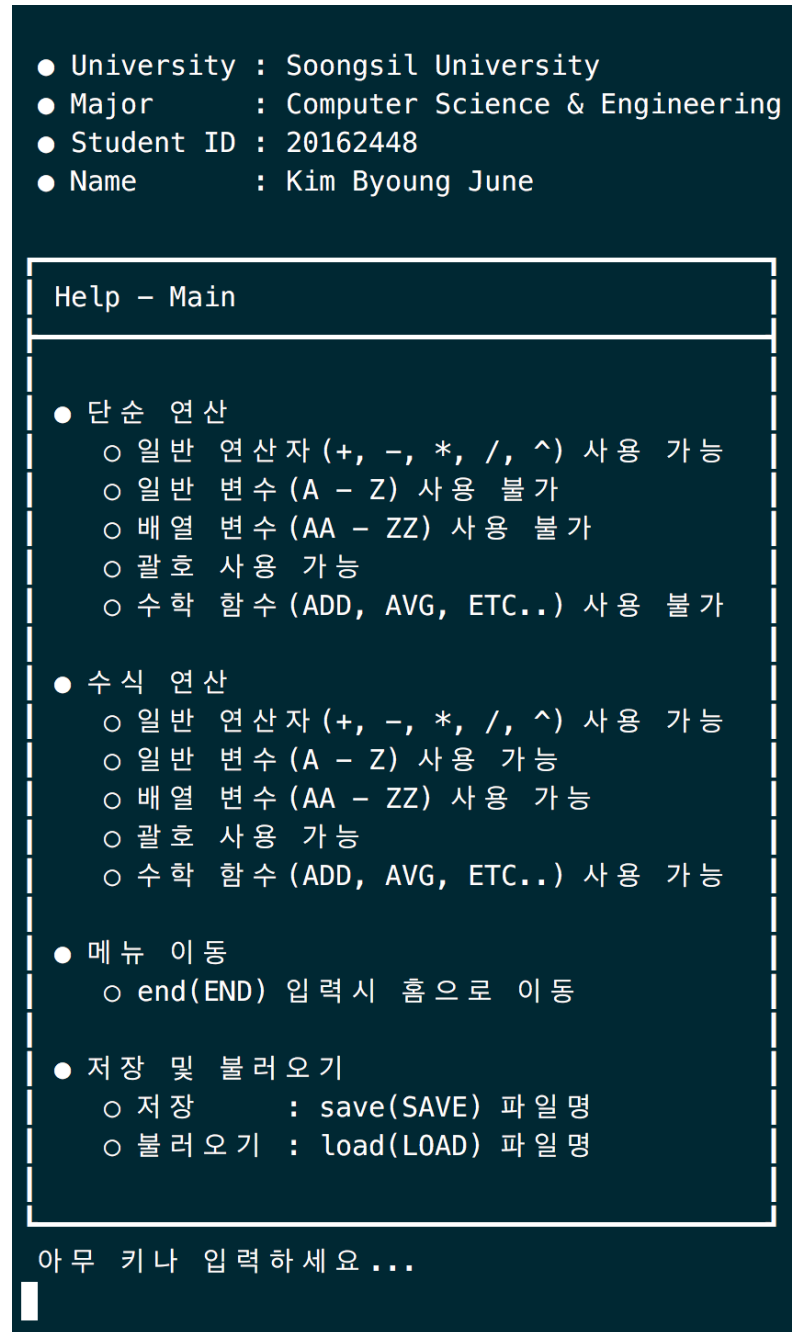
Formular Scientific Calculator - Main

입력 : A
ERROR : 변수 'A'가 선언되지 않음
아무 키나 입력하세요...
입력 : B
ERROR : 변수 'B'가 선언되지 않음
아무 키나 입력하세요...
입력 : load good
입력 : A
= 1
입력 : B
= 2
입력 : AA
= 1, 2, 3, 4, 5
입력 : BB
= 10, 20, 30, 40, 50
입력 : CC
= 1.1, 2.2, 3.3, 4.4, 5.5
입력 : █

```

〈그림 10〉단순 변수와 배열 변수 파일 저장 및 불러오기 기능

- 도움말 기능



〈그림 11〉 도움말 기능

V. 결론

지금까지 스택을 활용하여 후위표기법의 공학용 계산기를 제작해보았다. 스택은 알고 보면, 상당히 활용 할 곳이 많은 자료구조임을 알 수 있다. 본 연구의 계산기에서는 수식의 괄호 검사와 후위표기법이 대표적인 예이며, 후입선출의 성질을 이용하여 문자열에 대한 역순 문자열도 간단히 만들 수 있다. 또한 이러한 예를 운영체제(OS)에서도 찾아볼 수 있는데, 프로그램 간의 호출과 복귀에 따른 수행 순서를 보면 호출한 순서와 복귀하는 순서는 반대이기 때문에 가장 나중에 호출된 함수가 가장 먼저 실행을 완료하고 복귀한다. 따라서 함수의 호출과 복귀는 스택의 구조를 응용하여 관리할 수 있다.

우리는 이처럼 스택의 성질을 이용하여 기존의 복잡한 시간복잡도를 탈출할 수 있는 발판을 마련하였다. 단순히 공학용 계산기 제작에서 멈추지 않고, 한걸음 더 나아가 자기 참조 구조체(Linked List)와 같은 자료구조의 심화 학습을 통하여 데이터 관리 역량을 강화해야 할 것이다.