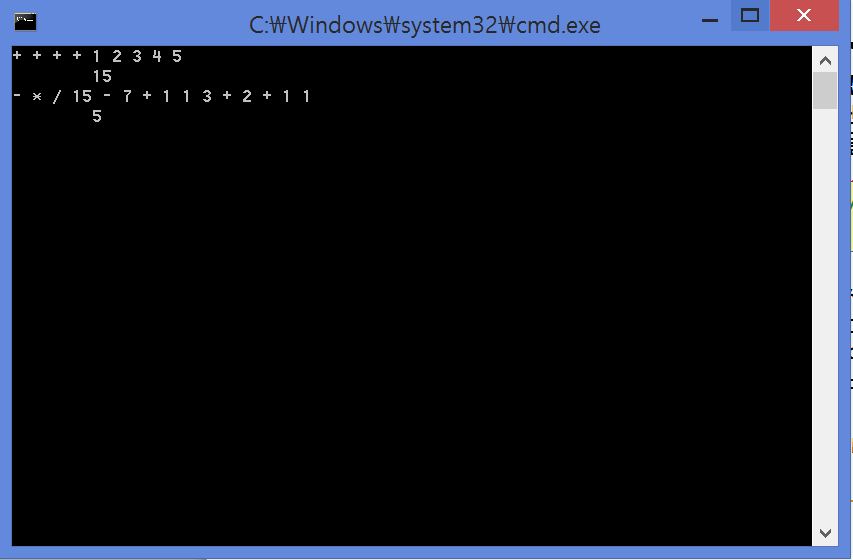
|  |  |
| --- | --- |
|  | **프로그래밍 개론Ⅰ**  **Lab09** |
| **분반** | **01** |
| **학번** | **1515655** |
| **이름** | **임소희** |
| **제출일** | **2015년 11월 11일** |

**과제**

**Polish prefix notation, Polish notation, or Prefix notation**

#실행 결과 (capture) 및 설명



전위 표기법은 연산기호를 앞에 놓고 뒤에 숫자를 배열하여 계산하는 방식이다.

위 캡쳐의 계산도 중위 표기법으로 표현하면 1+2+3+4+5=15가 되고,

(15/(7-(1+1)))\*3-(2+(1+1))=5가 된다.

후위 표기법에서 전위표기법으로 변환하기 위해서는 몇 가지만 수정하면 된다. Newline이 들어올 때와 default인 경우는 그대로 두고, number와 operator가 들어올 때만을 변경하면 된다.

우선 operator와 number를 구분하여 배열에 넣기 위해 각 각 opert[100], operand[100]으로 배열을 따로 둔다. 또한 연산기호와 숫자가 입력될 때, 일정 조건을 충족할 경우 연산을 수행하도록 opsp=-1라는 변수를 새로 둔다.

전위 표기법이므로 우선적으로 연산기호가 먼저 들어온다. +, -, \*, /가 들어올 시, 바로 계산하는 후위 표기법과 다르게 숫자가 아직 없으므로 계산 할 수 없다. 따라서 들어온 type(이 경우 연산기호)을 opert 배열에 저장하고 이 때 ++opsp를 증가시킨다.

Number가 들어오면 연산을 위한 for문이 충족되지 않아 switch에서 number와 연산기호를 수행못하므로 결국 operand[opsp]의 size()를 이용하여 크기를 재게 된다.opsp 순서로 저장하는 이유는 연산 시, 특정 조건 충족을 위해서이다.

Number가 입력됐을 때, 입력된 (숫자)문자를 숫자로 바꾸는 atof함수를 사용한다. 그 후 for문을 돌려 연산을 수행하는데, 이 때 for문의 조건은 연산기호 하나, 숫자 2개라는 최소의 조건을 충족 시 실행되어야 한다. 이 조건을 표현하기 위해서 opsp를 이용한다. 연산기호 하나 증가시, opsp=0이 된다.(연산기호 최소조건 충족) 또한 숫자가 2개 들어온 것을 확인해야 하므로 size()==operand[opsp]+2(숫자 최소조건 충족)하게 된다.

이 조건을 충족 후, opert[opsp]를 symbol이란 변수에 넣고, switch 함수에서 symbol이 각 기 사칙 연산일 때 case를 이용하여 연산한다. for문이 끝난 후 opsp—하여 주어진 연산이 끝나게 한다.