Homework 1B report

2014313303 홍태하(개발환경 : Linux)

1. 알고리즘

인풋이 postfix로 들어오기 때문에 연산자들간의 우선순위를 고려하지 않아도 된다. 숫자가 들어오면 stack에 push하여 저장해놓고 연산자가 들어오면 stack에서 두 개의 숫자를 pop하여 그 두 수를 연산한 후 결과 값을 다시 stack에 push하여 저장한다. 연산의 경우 먼저 pop된 수가 연산자의 뒤, 그 후에 pop된 수가 연산자의 앞에 오게 된다. (예를 들면 stack의 top에 2가 있고 그 바로 밑에 4가 있는 경우 / 연산을 진행하면 4/2가 되어 2라는 결과값을 얻게 된다.) 이 방식으로 모든 연산을 진행하고 stack에 남아있는 하나의 수가 결과값이 된다.

이 문제의 경우 네 자릿수의 8진수로 인풋이 들어오기 때문에 십진수로 바꾸어 stack에 저장하고 계산도 십진수로 하였다. + 연산의 경우 8진수의 네 번째 자리를 넘어갈 수 있지만 넘어가더라도 8진수의 다섯 번째 자리 값이 1을 초과하지 않기 때문에 넘는 경우 8^4을 빼주어서 truncation을 하였다. \* 연산의 경우 8진수의 다섯 번째 자리 값이 1을 초과할 수 있기 때문에 8진수로 바꾸어서 truncation을 하고 다시 십진수로 바꾸었다. 이렇게 모든 계산을 한 후 마지막 결과값을 네 자리 8진수로 바꾸어 출력하였다.

1. 코드 설명.
2. 인풋 받기.

먼저 input\_c라는 변수를 통해서 char형으로 인풋을 하나씩 받아들인다. 인풋의 종류는 4가지가 있는데 0부터 7까지의 숫자, 5가지 종류의 연산자, 스페이스, 엔터가 그것이다.

엔터가 들어왔을 경우 인풋의 끝이므로 인풋을 더 이상 받아들이지 않는다. 스페이스의 경우 단지 숫자나 연산자들간의 구분을 위한 것이므로 아무 작업도 하지 않고 다시 인풋을 받아들이면 된다.

0부터 7까지의 숫자가 들어온 경우 3개의 숫자를 더 받아야 하고 이 숫자들을 10진수로 바꾸어서 operand라는 이름의 스택에 push해야 한다. 따라서 input\_i라는 변수에 input\_c를 int형으로 바꾼 값을 넣어준다. 8진수의 자릿값에 맞게 8을 거듭제곱 해주고 그 값들을 input이라는 변수에 더해준다. 4 자릿수의 모든 수들이 더해지면 input을 operand스택에 push해준다. Input은 0으로 초기화하고 다시 새로운 인풋을 받는다.

연산자가 인풋으로 들어온 경우 계산을 진행한다. 계산은 Calc라는 이름의 함수를 만들어서 진행한다.

1. Calc 함수

인풋으로 받은 연산자에 따라 연산을 진행하는 함수이다. Ope1에는 연산자 앞에 올 피연산자를 저장하고 ope2에는 연산자 뒤에 올 피연산자를 저장한다. Stack의 가장 위에 있는 수가 ope2가 되고 그 밑에 수가 ope1이 된다. Peek함수를 이용해서 피연산자들을 저장하고 Pop해버린다. 그 후 연산을 진행하고 그 결과값을 변수 result에 저장해서 stack에 push한다. +의 경우 8진수의 다섯 번째 자릿수로 넘어가는 경우가 발생하지만 그 경우 8^4를 빼주면 된다. 하지만 \*의 경우 8진수의 다섯 번째 자릿수로 어느 정도가 넘어갈 지 알 수 없으므로 truncation이라는 이름의 함수를 만들어서 처리해준다.

1. Truncation 함수

oct라는 이름의 4칸짜리 배열을 생성해서 8진수를 저장한다. Oct[3]에 가장 작은 자릿수의 값이 들어가고 oct[0]에 가장 큰 자릿수의 값이 들어간다. 그렇게 4칸을 채워서 그 값을 취하면 8진수의 다섯 번째 자리를 넘는 값은 없어진다. 이제 oct배열에 있는 값들을 다시 십진수로 바꾸어서 리턴한다.

1. 마무리

모든 연산이 끝나면 stack에 하나 남아있는 값이 결과값이다. 하지만 십진수이므로 8진수로 바꾸어서 출력한다.

1. 예시

1002 0110 \* 2000 / 가 인풋으로 들어온 경우, 1002를 십진수 514로 바꾸어 stack에 push하고 0110도 십진수 72로 바꾸어 stack에 push한다. \*가 나왔으므로 514와 72를 꺼내서 514\*72를 진행한다. 514\*72는 37008인데 8^4를 넘어가므로 truncation함수에 넣는다. 37008을 8진수로 바꾸면 110220인데 뒤에 4자리만 저장하므로 0220이 된다. 0220을 다시 십진수로 바꾸면 144이므로 144를 stack에 넣는다. 2000을 십진수 1024로 바꾸어 stack에 push하고 /가 나왔으므로 144과 1024를 꺼내서 144/1024를 진행한다. 결과값 0을 stack에 넣고 인풋이 끝났으므로 stack에 남은 수를 꺼내서 8진수로 바꾸어 출력하면 0000이 된다.