자료구조응용

09. 후위표기법 (10점)

2020.4.13. (월)

1. 후위표기법(postfix notation)으로 표현된 하나의 수식을 파일로 입력받아 그 계산결과를 화면에 출력하는 프로그램을 작성하라. (5점)

[프로그램 설명]

입력파일(input.txt) : 62/3-42*+

사용되는 연산자 : +, -, *, /, %
사용되는 피연산자 : 1~9 사이의 한 자리 정수
'(', ')' 연산자는 입력되지 않음
divide by zero에 대한 테스트 및 처리는 구현하지 않음

```
int stack[MAX_STACK_SIZE];
int top = -1;
```

```
typedef enum {lparen, rparen, plus, minus, times, divide,
              mod, eos, operand} precedence;
```

```
int eval(void)
{/* evaluate a postfix expression, expr, maintained as a
    global variable. '\0' is the the end of the expression.
    The stack and top of the stack are global variables.
    getToken is used to return the token type and
    the character symbol. Operands are assumed to be single
    character digits */
  precedence token;
  char symbol;
  int op1, op2;
  int n = 0; /* counter for the expression string */
  top = -1;
  token = getToken(&symbol, &n);
  while (token != eos) {
    if (token == operand)
      push(symbol-'0'); /* stack insert */
    else {
      /* pop two operands, perform operation, and
         push result to the stack */
      op2 = pop(); /* stack delete */
      op1 = pop();
      switch(token) {
        case plus: push(op1+op2);
                   break;
        case minus: push(op1-op2);
                    break;
        case times: push(op1*op2);
                    break;
        case divide: push(op1/op2);
                     break;
        case mod: push(op1%op2);
      }
    }
    token = getToken(&symbol, &n);
  }
  return pop(); /* return result */
}
```

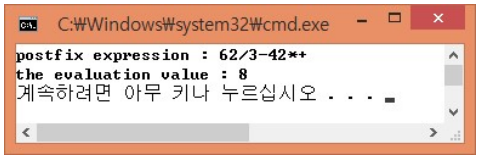
Program 3.13: Function to evaluate a postfix expression

```
precedence getToken(char *symbol, int *n)
{/* get the next token, symbol is the character
    representation, which is returned, the token is
    represented by its enumerated value, which
    is returned in the function name */
   *symbol = expr[(*n)++];
   switch (*symbol) {
      case '(' : return lparen;
      case ')' : return rparen;
      case '+' : return plus;
      case '-' : return minus;
      case '/' : return divide;
      case '*' : return times;
      case '%' : return mod;
      case '\0' : return eos;
      default  : return operand; /* no error checking,
                                    default is operand */
   }
}
```

Program 3.14: Function to get a token from the input string

※ '(', ')'의 경우 본 문제에서는 사용되지 않음

[실행 예]



2. 중위표기법(infix notation)으로 표현된 하나의 수식을 파일로 입력받아 후위표기법(postfix notation)으로 변환하여 화면 및 파일에 동시에 출력하는 프로그램을 작성하라. (5점)

[프로그램 설명]

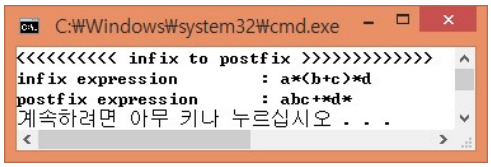
입력파일("input.txt") : a*(b+c)*d [or (4/(2-2+3))*(3-4)*2 or (4/(a-b+3))*(c-4)*2] 화면출력 & 파일출력("output.txt") : abc+*d* ※ 입력수식의 문자열 길이는 최대 80으로 함
사용되는 연산자 : +, -, *, /, %, (,) 사용되는 피연산자 : 알파벳 소문자, 1~9 사이의 한 자리 정수 ※ 피연산자가 모두 1~9의 한 자리 정수면, 출력결과를 1번 문제의 입력으로 사용 가능
posfix(), printToken() 함수의 화면출력 부분에 파일출력 추가 void printToken(precedence); 직접 구현

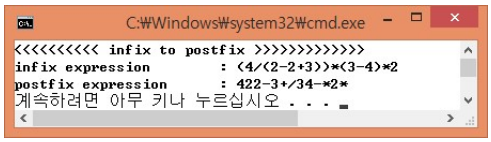
```
typedef enum {lparen, rparen, plus, minus, times, divide,
                        mod, eos, operand} precedence;

/* isp and icp arrays -- index is value of precedence
   lparen, rparen, plus, minus, times, divide, mod, eos */
int isp[] = {0,19,12,12,13,13,13,0};
int icp[] = {20,19,12,12,13,13,13,0};

precedence stack[MAX_STACK_SIZE];

top = -1;
```

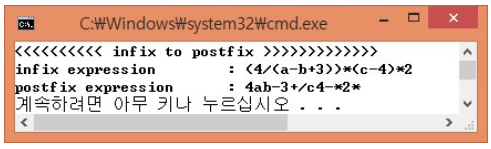
```
void postfix(void)
{/* output the postfix of the expression. The expression
    string, the stack, and top are global */
  char symbol;
  precedence token;
  int n = 0;
  top = 0;    /* place eos on stack */
  stack[0] = eos;
  for (token = getToken(&symbol, &n); token != eos;
                          token = getToken(&symbol,&n)) {
    if (token == operand)
      printf("%c", symbol);
    else if (token == rparen) {
      /* unstack tokens until left parenthesis */
      while (stack[top] != lparen)
        printToken(pop());
      pop();  /* discard the left parenthesis */
    }
    else {
      /* remove and print symbols whose isp is greater
         than or equal to the current token's icp */
      while(isp[stack[top]] >= icp[token])
        printToken(pop());
      push(token);
    }
  }
  while ( (token = pop()) != eos)
    printToken(token);
  printf("\n");
}
```

Program 3.15: Function to convert from infix to postfix

[실행 예]







■ 제출 형식

- 솔루션 이름　: DS 09
- 프로젝트 이름 : 1, 2
- 각 소스파일에 주석처리
  “학번 이름”
  “본인은 이 소스파일을 복사 없이 직접 작성하였습니다.”

- 솔루션 정리 후 솔루션 폴더를 “학번.zip”으로 압축하여 과제 게시판에 제출

■ **주의**

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!

- 마감 : 4월 14일(화) 자정