|  |  |
| --- | --- |
|  | **프로그래밍 개론Ⅰ**  **Lab08** |
| **분반** | **01** |
| **학번** | **1515655** |
| **이름** | **임소희** |
| **제출일** | **2015년 10월 30일** |

**과제 1**

**strindex(source[], start, pattern[])**

#실행 결과 (capture) 및 설명



문장에서 특정 성분을 추출하여 개수를 세는 프로그램이다.strindex 함수를 만든다.

문장 s[]를 char로 선언하고, 추출할 성분도 char t[]로 선언한다. 또 문장 내 추출 성분을 검사하는데 있어 문장 전체 길이가 되는 start도 int(글자 ‘개수’를 세야하므로)로 선언한다. 그러므로 start는 index+1이 된다.(마지막 비트는 ‘\0’으로 채워진다.)

S[start]가 ‘\0’을 만나기 전까지 계속 실행되며, 실행마다 start를 1씩 증가시킨다(글자 1개씩 읽으므로). J를 시작점, k=0으로 초기값을 두고 t[k]가 null문자를 만나기 전까지, 그리고 s[j]와 t[k]가 동일한 동안 수행한다. 즉, 원하는 성분이 문장 내에 있는 동안 수행한다. K가 양수이고 t[k]가 null문자를 만나면 start를 초기화시킨다. 예를 들어 the가 원하는 성분이면 문장 내에 ‘e’가 나오면 실행을 끝내고, 문장 내 the 발견 개수 1을 증가시킨다. 그리고 초기화하여 다음 문장 성분에서 원하는 성분을 찾는 작업을 문장이 끝날 때 까지 반복한다.)

만약 원하는 성분이 문장 내에 존재하지 않으면 -1을 리턴한다.

**과제 2**

**멱승 연산자(^)**

#실행 결과 (capture). 및 설명

계산기 프로그램이다. 우선 main 함수에서 수와 식을 넣어야 한다. 이 때 입력한 수는 정수가 아닌 문자(char)이므로 숫자로 바꿔주는 작업을 한다(atof(s)를 이용함). 그 다음 사칙 연산 계산 시, 어떻게 계산 되어야 하는지 적어나간다. 이 때 사용하는 것이 stack이다. 따라서 stack함수를 만들어야 한다. Stack이 들어갈 수 있는(pop) 범위를 설정하고, 초과시 error 메시지를 띄운다. Stack에 있는 것을 꺼낼 때(pop), 스택에 수나 연산이 존재할 경우 꺼내서 사용하고, 없으면 error 메시지메 띄어 stack이 비웠다고 정한다. 이 때, 출력 시 소수 첫째 자리까지 나오도록 return 0.0을 적는다. 계산할 식을 적을 시, 숫자와 부호 이외에는 계산이 불가능하도록 getop 함수를 만든다. 숫자와 ‘.’일 때는 계산기가 정상 작동하도록 하고, 나머지 문자가 들어오면 error 표시를 띄우도록 한다.

사칙 연산을 적을 시, ‘+’와 ‘\*’는 연산 순서가 크게 상관 없으므로, stack에서 push해서 각 각 pop하여 +,\* 연산을 한다. (push(pop() + pop()),push(pop() \* pop()))

그러나 ‘-‘와 ‘/’는 연산 순서가 중요하다. –인 경우, 먼저 뽑은 것이 후에 계산 되도록 한다. 예를 들어 op2를 미리 뽑고, 연산 시, pop을 하나 한 것에 op2를 빼주는 것이다.(stack은 나중에 넣은 것부터 먼저 뽑히므로 일반 계산 순서와 반대로 해야 한다)./의 경우도 op2를 우선 뽑고(이 때 op2는 0이 반드시 0이 아니어야 한다.) pop을 하여 그 것을 나누어 준다. Pop()/op2 형태로 나눠야 하는 것이다.

멱등 연산자를 구할 경우, d=1을 선언한다. 그리고 op2를 먼저 구하고, op1도 pop한다.‘op1의 op2승’의 형태가 되어야 하므로, op2가 0이 될 때까지 d=op1\*d를 반복한다. 마지막으로 enter를 눌러 답을 출력한다.