

Lab#4 STACK practice

Write a program to read a text file and print whether or not a line is a palindrome.

1. palindrome: A palindrome is a string that reads the same forward and backward.

즉 앞에서부터,... 뒤로부터 읽은 것이 같은 string임

(ex) radar 0 1 00 11 aba 1101011 등

2. Testing first data file: abccba abckcba abbc abbacd

correct output:

abccba	a palindrome
abckcba	a palindrome
abbc	not a palindrome
abbacd	not a palindrome

3. Testing second data file: (PUSH only parentheses!!)

=> save the parentheses into Buffer.

{ [A + B] - [(C - D)] }	balanced
(A * B) + (C * D)	balanced
((A + B)	unbalanced
A + B (unbalanced

4. Algorithm

```
while (infile.getline(buffer,80)) { //한 line씩 처리함
```

```
    len = strlen(buffer); //string의 길이, 글자수.
```

```
    if len 이 짝수이면 { // (len % 2) == 0, if len is EVEN
```

```
        while (i < (len / 2)) //length의 반만큼 PUSH. 'abccba'의 경우 push a, push b, push c.
```

```
            push(buffer[i]);
```

```
    }
```

```
    else if len == 홀수 이면 { // (len % 2) != 0, if len is ODD
```

```
        while (i < (len / 2)) //length의 반만큼 PUSH. 'abckcba'의 경우 push a, push b, push c.
```

```
            push(buffer[i]); // 이젠 buffer 에 kcba 남았음
```

```
            I++; // k를 건너 뛴다.
```

```
    }
```

```

while (buffer[i] != '\0') {
    compare (buffer[i] and pop(ch))    // if not SAME, then BREAK;
    //만일 같으면 다음 글자 비교, 다르면 error message & break
    If (buffer[i] = "empty") then print "PALINDROME"
}

```

Extra Point:

4-1. Circular Queue (Problem: Implement CIRCULAR Queue ADT)

```
#define Queue_Size 5
```

```

while (1) {
    print ("command: enqueue, dequeue, print, quit: ");
    get input;
    switch (input[0]) {
        'e' : if (!queue_full())    eneueue();
                else                printf("Queue is full\n");    break;
        'd' : if (!queue_empty())  dequeue();
                else                printf("Queue is empty\n");    break;
        'p' : if (!queue_empty())  printqueue();
                else                printf("Queue is empty\n");    break;
        'e' : exit(0);
        default: printf("Bad Command\n");    }
}

```

* 검사: (**enqueue, dequeue, queue_full, queue_empty**의 네가지 함수 검사를 위해서)

- 1) enqueue 'a', enqueue 'b', enqueue 'c', enqueue 'd', enqueue 'e' --> '**Queue Full**'출력
- 2) print ==> **a,b,c,d** 출력
- 3) dequeue, dequeue, dequeue, dequeue, dequeue => '**Queue Empty**' 출력