#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include <stdio.h>

#define TRUE 1

#define FALSE 0

#define MAX\_STRING 100

#define MAX\_STACK\_SIZE 100

**typedef char element; // 수정: char형의 문자를 스택에 넣는다**

typedef struct {

element stack[MAX\_STACK\_SIZE];

int top;

} StackType;

// 스택 초기화 함수

void init(StackType \*s)

{

s->top = -1;

}

// 공백 상태 검출 함수

int is\_empty(StackType \*s)

{

return (s->top == -1);

}

// 포화 상태 검출 함수

int is\_full(StackType \*s)

{

return (s->top == (MAX\_STACK\_SIZE-1));

}

// 삽입함수

void push(StackType \*s, element item)

{

if( is\_full(s) ) {

fprintf(stderr,"스택 포화 에러\n");

return;

}

else s->stack[++(s->top)] = item;

}

// 삭제함수

element pop(StackType \*s)

{

if( is\_empty(s) ) {

fprintf(stderr, "스택 공백 에러\n");

exit(1);

}

else return s->stack[(s->top)--];

}

// 피크함수

element peek(StackType \*s)

{

if( is\_empty(s) ) {

fprintf(stderr, "스택 공백 에러\n");

exit(1);

}

else return s->stack[s->top];

}

int palindrome(char str[])

{

StackType s; int i;

char ch, chFromStack;

int len = strlen(str);

// 스택을 초기화하라

init(&s);

//str의 문자들을 스택에 넣는다

for (i = 0; i < len; i++) {

ch = str[i];

push(&s, ch);

}

//스택에서 하나씩 빼면서 str의 문자들과 차례로 비교

for (i = 0; i < len; i++) {

// 코드 작성

}

return TRUE; //palindrome임

}

int main(void)

{

char word[MAX\_STRING];

printf("Enter a word to check palindrome: ");

scanf("%s", word);

if (palindrome(word))

printf("palindrome입니다.\n");

else

printf("palindrome이 아닙니다.\n");

}