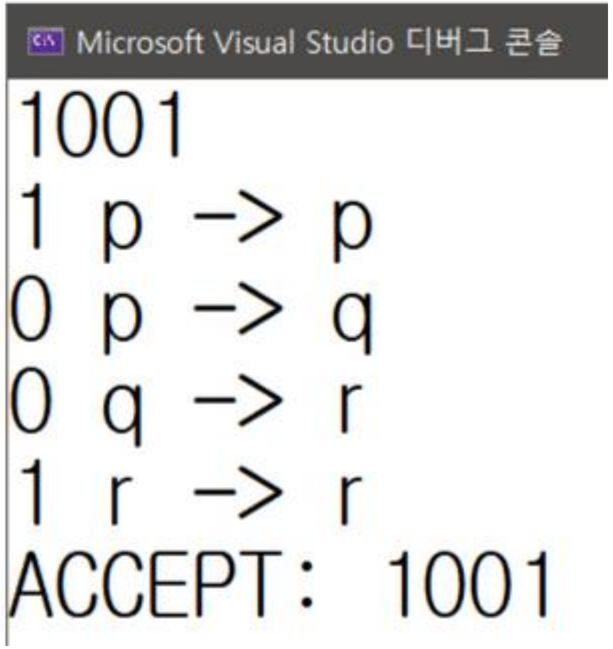
DFA 결과 발표

1891179 김현학

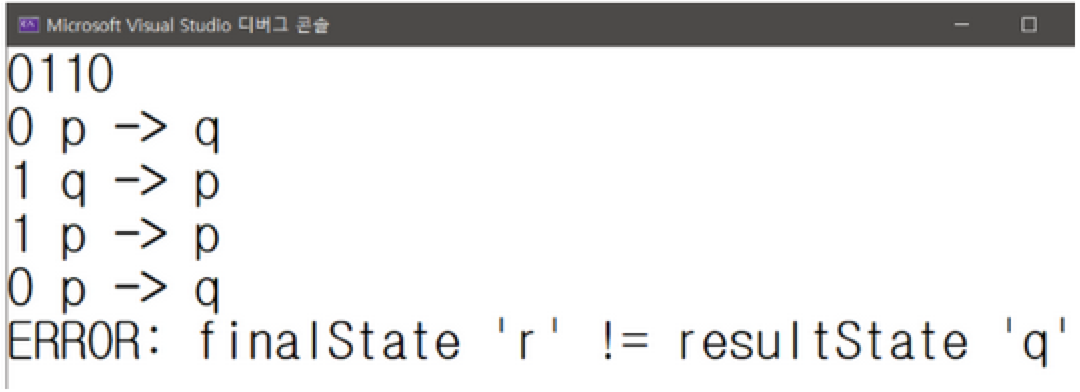
결과

입력: 1001 (Debug 모드)



```
1001
1 p -> p
0 p -> q
0 q -> r
1 r -> r
ACCEPT: 1001
```

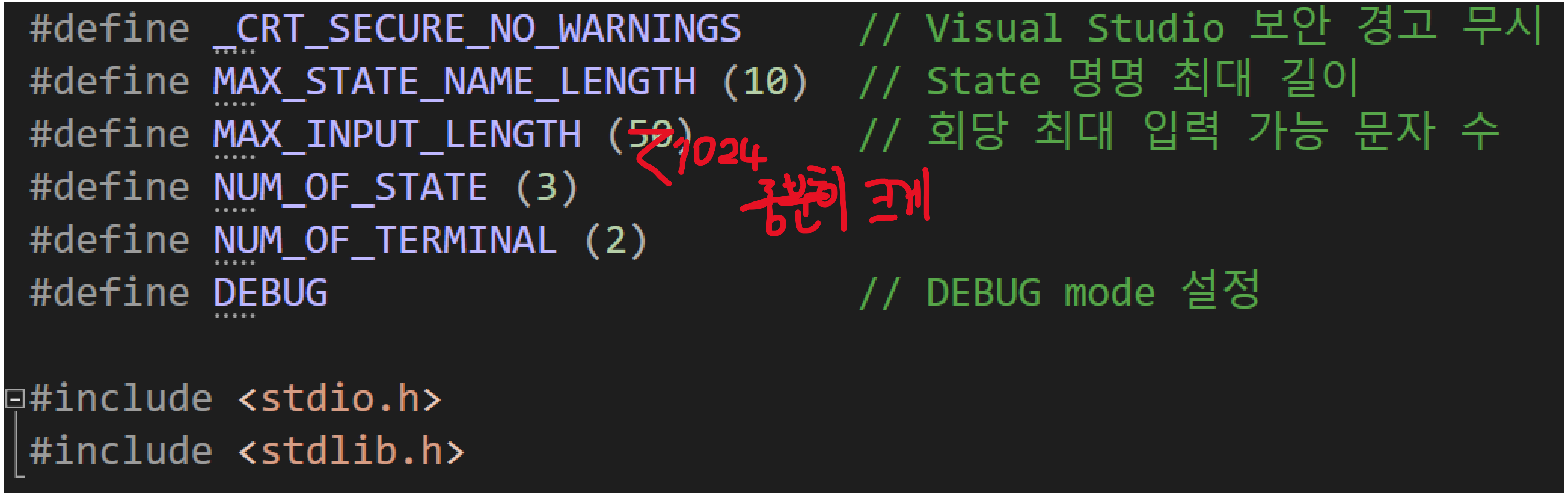
입력: 0110 (Debug 모드)



```
0110
0 p -> q
1 q -> p
1 p -> p
0 p -> q
ERROR: finalState 'r' != resultState 'q'
```

전처리기 및 매크로, 헤더파일

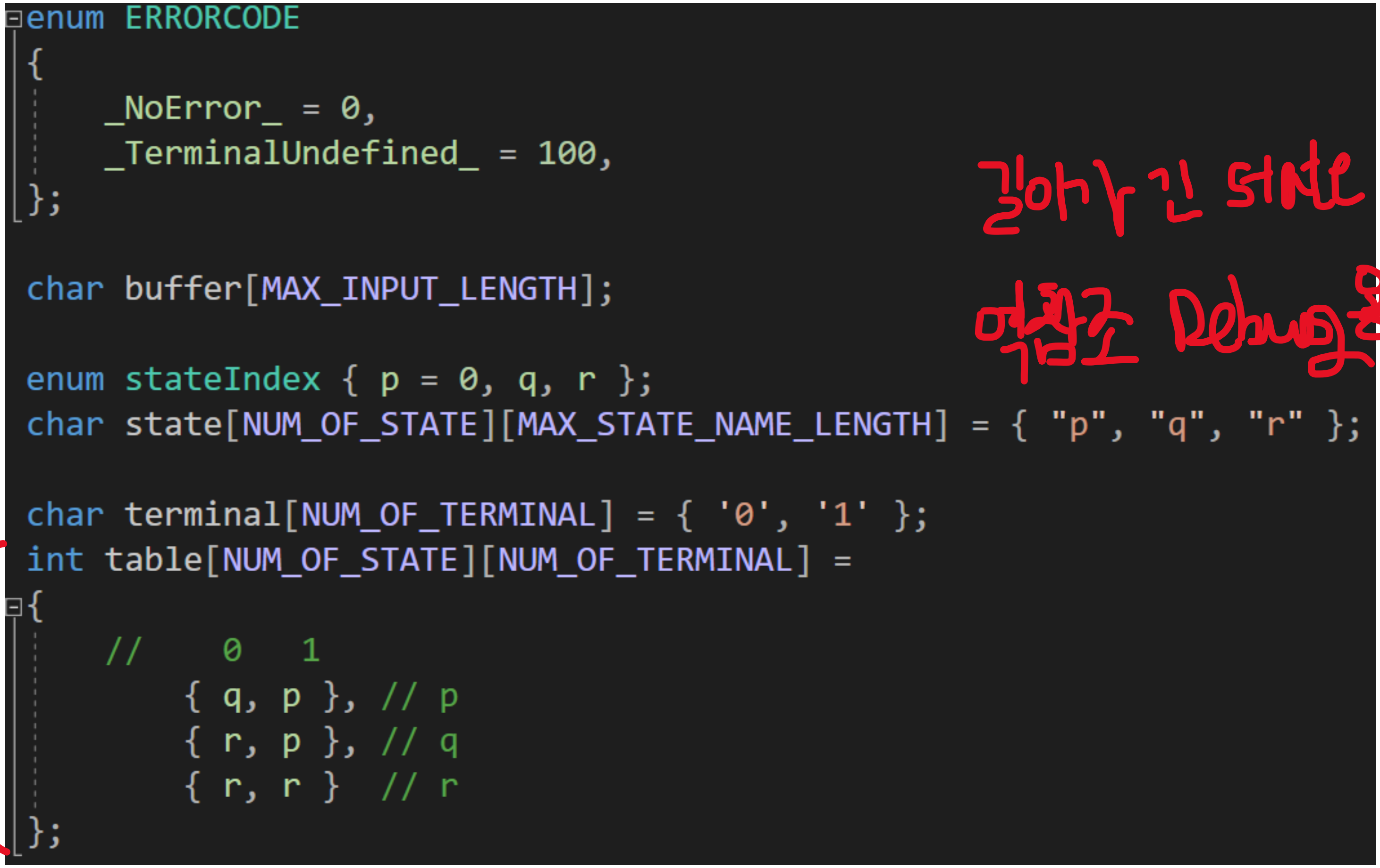
C언어



```
#define _CRT_SECURE_NO_WARNINGS      // Visual Studio 보안 경고 무시
#define MAX_STATE_NAME_LENGTH (10)   // State 명명 최대 길이
#define MAX_INPUT_LENGTH (50)        // 회당 최대 입력 가능 문자 수
#define NUM_OF_STATE (3)
#define NUM_OF_TERMINAL (2)
#define DEBUG                        // DEBUG mode 설정

#include <stdio.h>
#include <stdlib.h>
```

<1024
충분히 크게

전역 변수

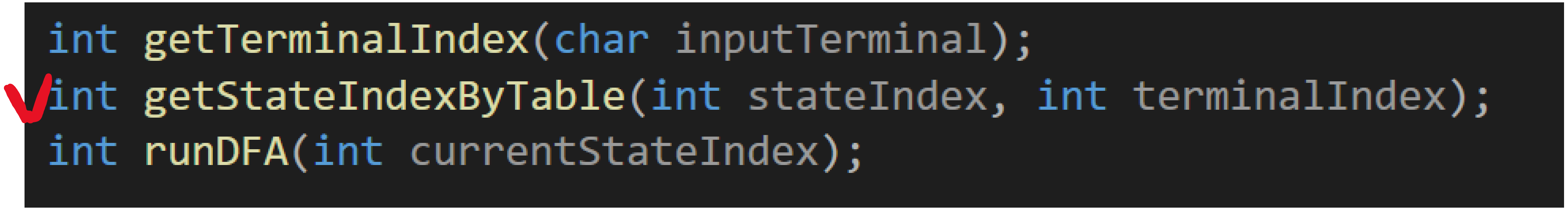


```
enum ERRORCODE
{
    _NoError_ = 0,
    _TerminalUndefined_ = 100,
};

char buffer[MAX_INPUT_LENGTH];

enum stateIndex { p = 0, q, r };
char state[NUM_OF_STATE][MAX_STATE_NAME_LENGTH] = { "p", "q", "r" };

char terminal[NUM_OF_TERMINAL] = { '0', '1' };
int table[NUM_OF_STATE][NUM_OF_TERMINAL] =
{
    //    0   1
        { q, p }, // p
        { r, p }, // q
        { r, r }  // r
};
```

길어지긴 State 고려

역참조 Debug용

함수



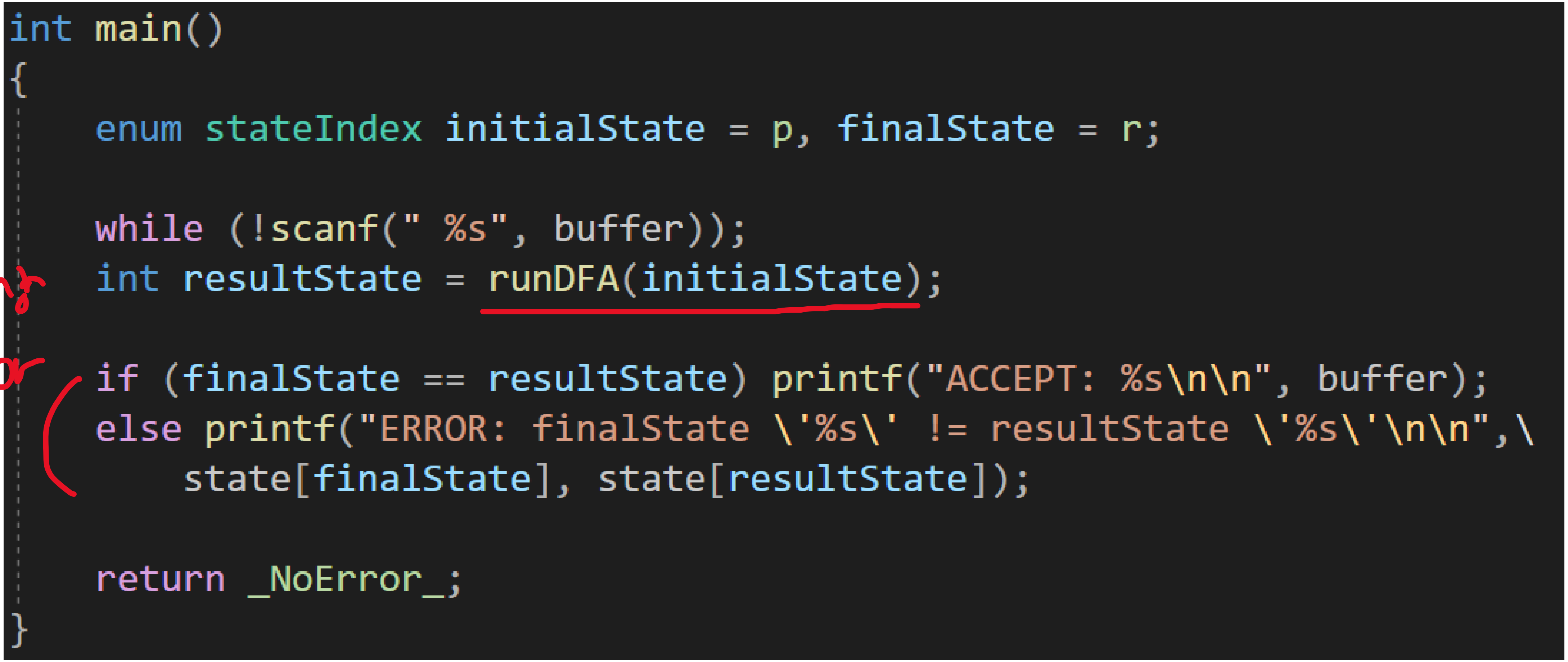
```
int getTerminalIndex(char inputTerminal);
int getStateIndexByTable(int stateIndex, int terminalIndex);
int runDFA(int currentStateIndex);
```

차이
H/T

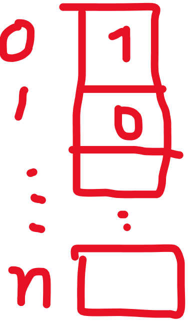
Main 함수

Final
2개 이상

for

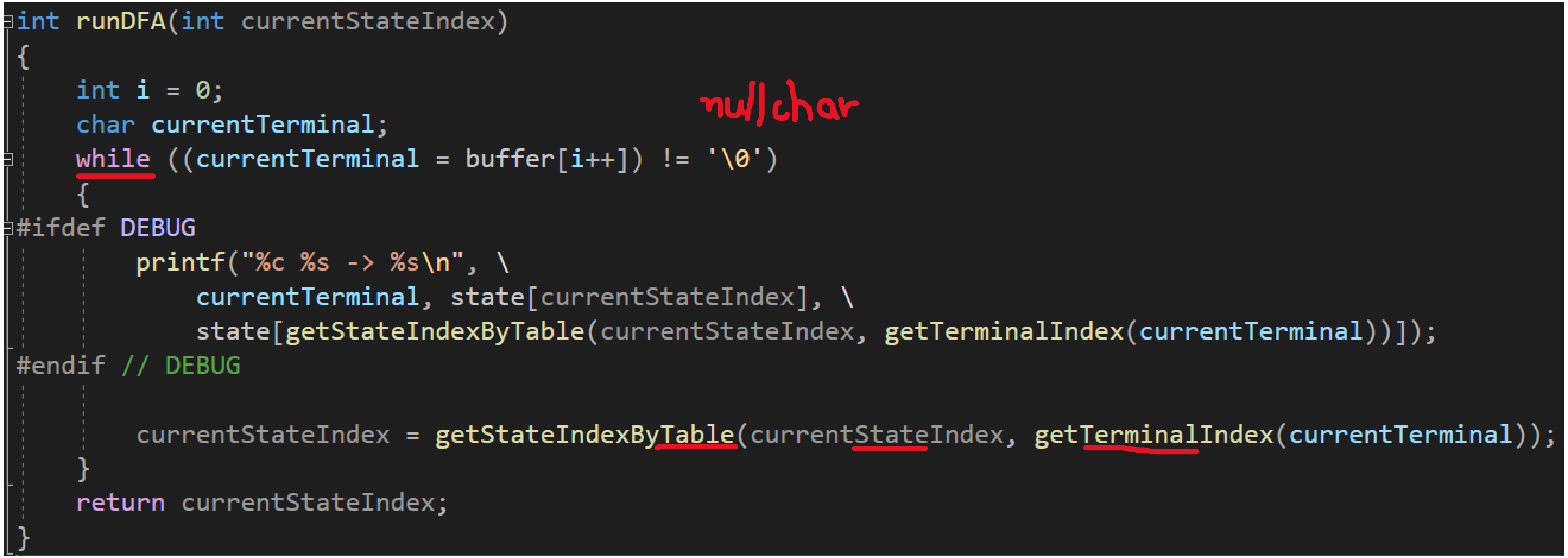


```
int main()
{
    enum stateIndex initialState = p, finalState = r;

    while (!scanf(" %s", buffer));
    int resultState = runDFA(initialState);

    if (finalState == resultState) printf("ACCEPT: %s\n\n", buffer);
    else printf("ERROR: finalState \'%s\' != resultState \'%s\'\n\n",\
        state[finalState], state[resultState]);

    return _NoError_;
}
```



isFinal

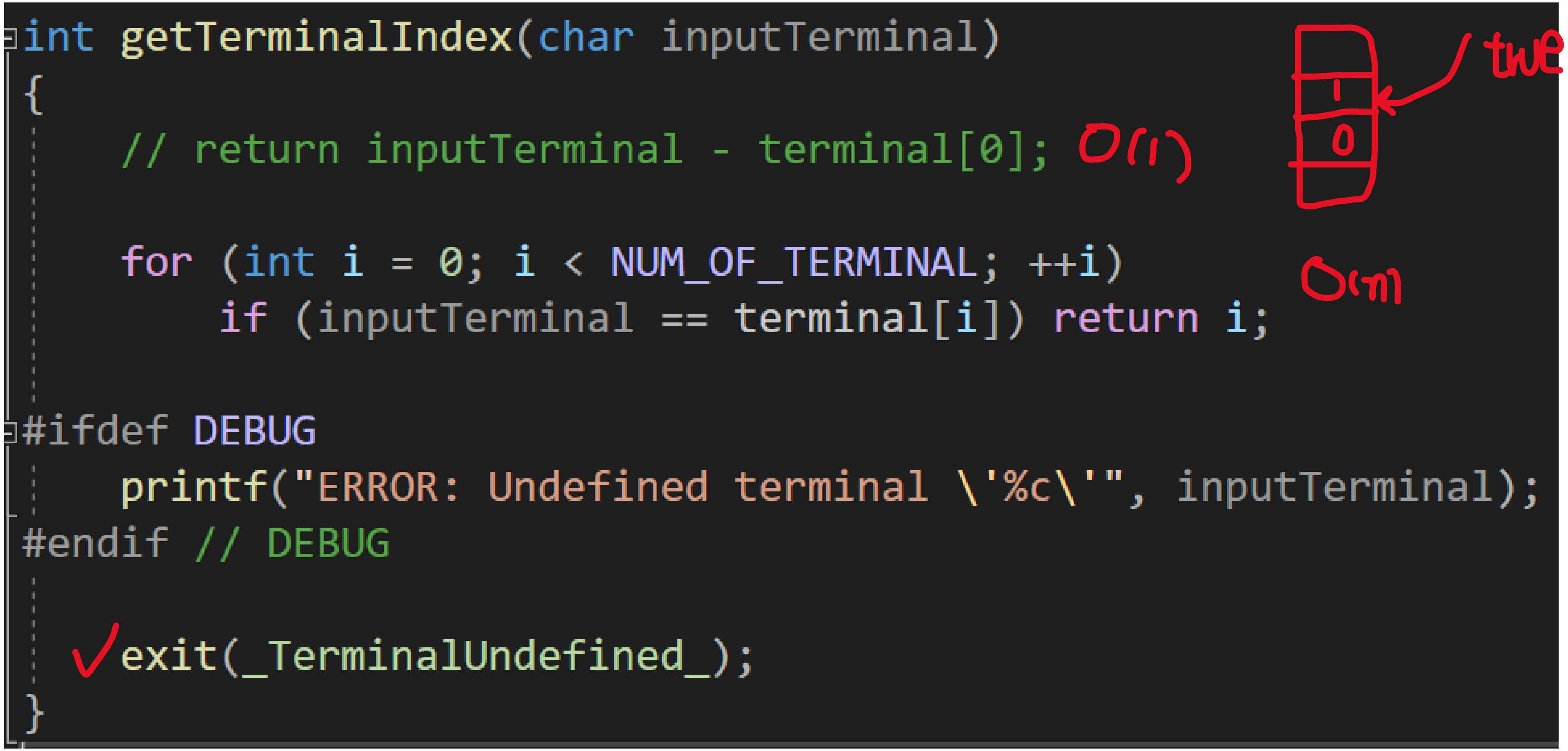
runDFA



```
int runDFA(int currentStateIndex)
{
    int i = 0;
    char currentTerminal;
    while ((currentTerminal = buffer[i++]) != '\0')
    {
#ifdef DEBUG
        printf("%c %s -> %s\n", \
            currentTerminal, state[currentStateIndex], \
            state[getStateIndexByTable(currentStateIndex, getTerminalIndex(currentTerminal))]);
#endif // DEBUG

        currentStateIndex = getStateIndexByTable(currentStateIndex, getTerminalIndex(currentTerminal));
    }
    return currentStateIndex;
}
```

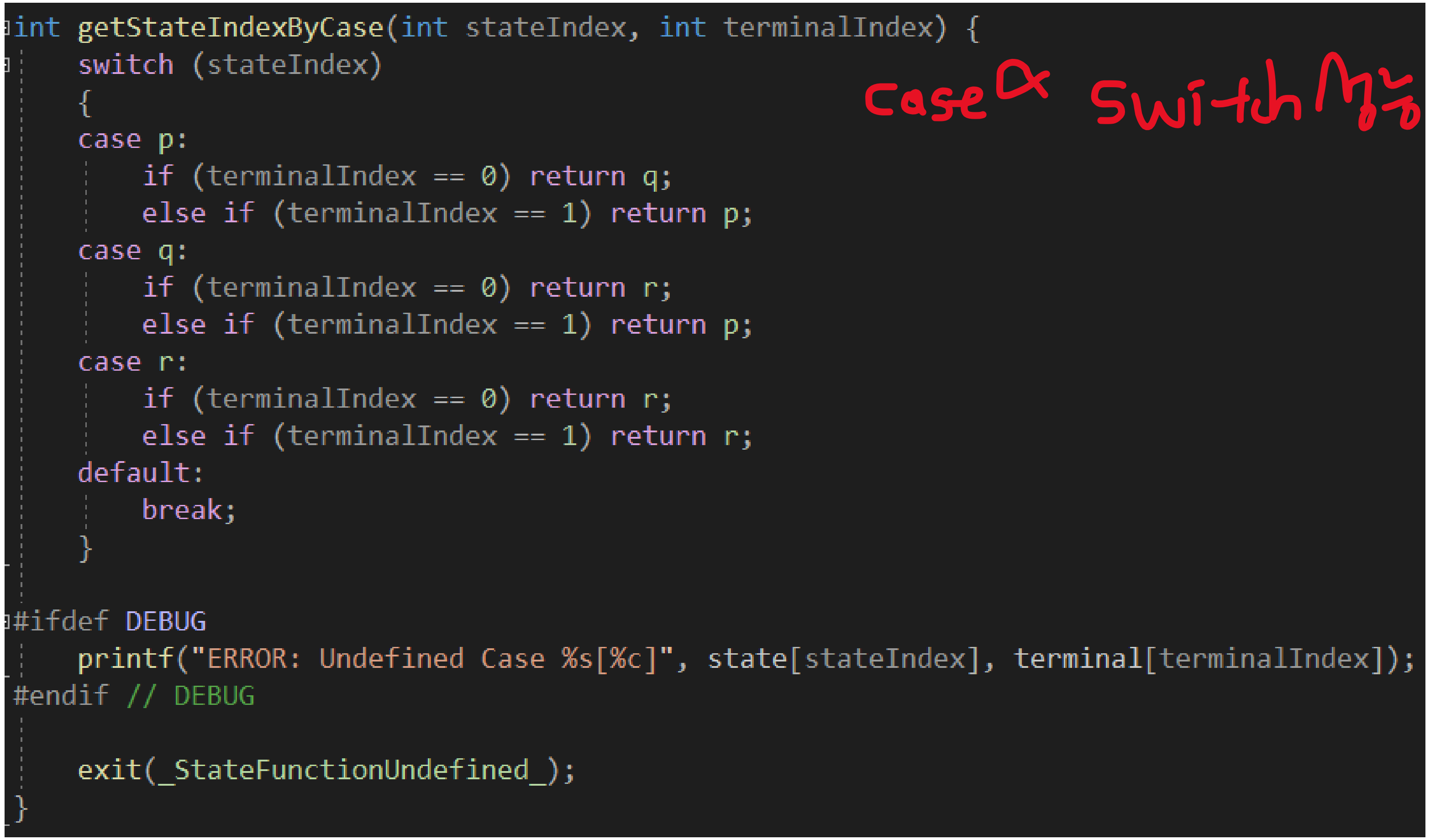
nullchar

GetTerminalIndex



```
int getTerminalIndex(char inputTerminal)
{
    // return inputTerminal - terminal[0];

    for (int i = 0; i < NUM_OF_TERMINAL; ++i)
        if (inputTerminal == terminal[i]) return i;

#ifdef DEBUG
    printf("ERROR: Undefined terminal \'%c\'", inputTerminal);
#endif // DEBUG

    exit(_TerminalUndefined_);
}
```

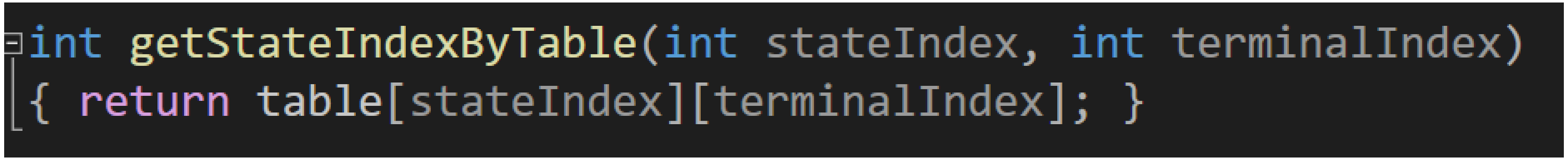
Hard-wired-dfa GetStateIndexByTable



```
int getStateIndexByCase(int stateIndex, int terminalIndex) {
    switch (stateIndex)
    {
    case p:
        if (terminalIndex == 0) return q;
        else if (terminalIndex == 1) return p;
    case q:
        if (terminalIndex == 0) return r;
        else if (terminalIndex == 1) return p;
    case r:
        if (terminalIndex == 0) return r;
        else if (terminalIndex == 1) return r;
    default:
        break;
    }

#ifdef DEBUG
    printf("ERROR: Undefined Case %s[%c]", state[stateIndex], terminal[terminalIndex]);
#endif // DEBUG

    exit(_StateFunctionUndefined_);
}
```

case & switch [illegible]

Table-driven-dfa GetStateIndexByTable



```
int getStateIndexByTable(int stateIndex, int terminalIndex)
{ return table[stateIndex][terminalIndex]; }
```