

<컴파일러 과제 2 : 전이 다이어그램 구현>



제출일	2022.04.22	전공	소프트웨어학과
과목	컴파일러	학번	2020039096
담당교수	이재성 교수님	이름	백인혁

목차

1. 문제 분석
2. 전이 다이어그램 작성
3. 코드 설명
4. 실행 예시

1. 문제 분석

주어진 정규표현에 대한 전이 다이어그램을 작성하고 전이다이어그램을 구현하는 것이 목표이다.

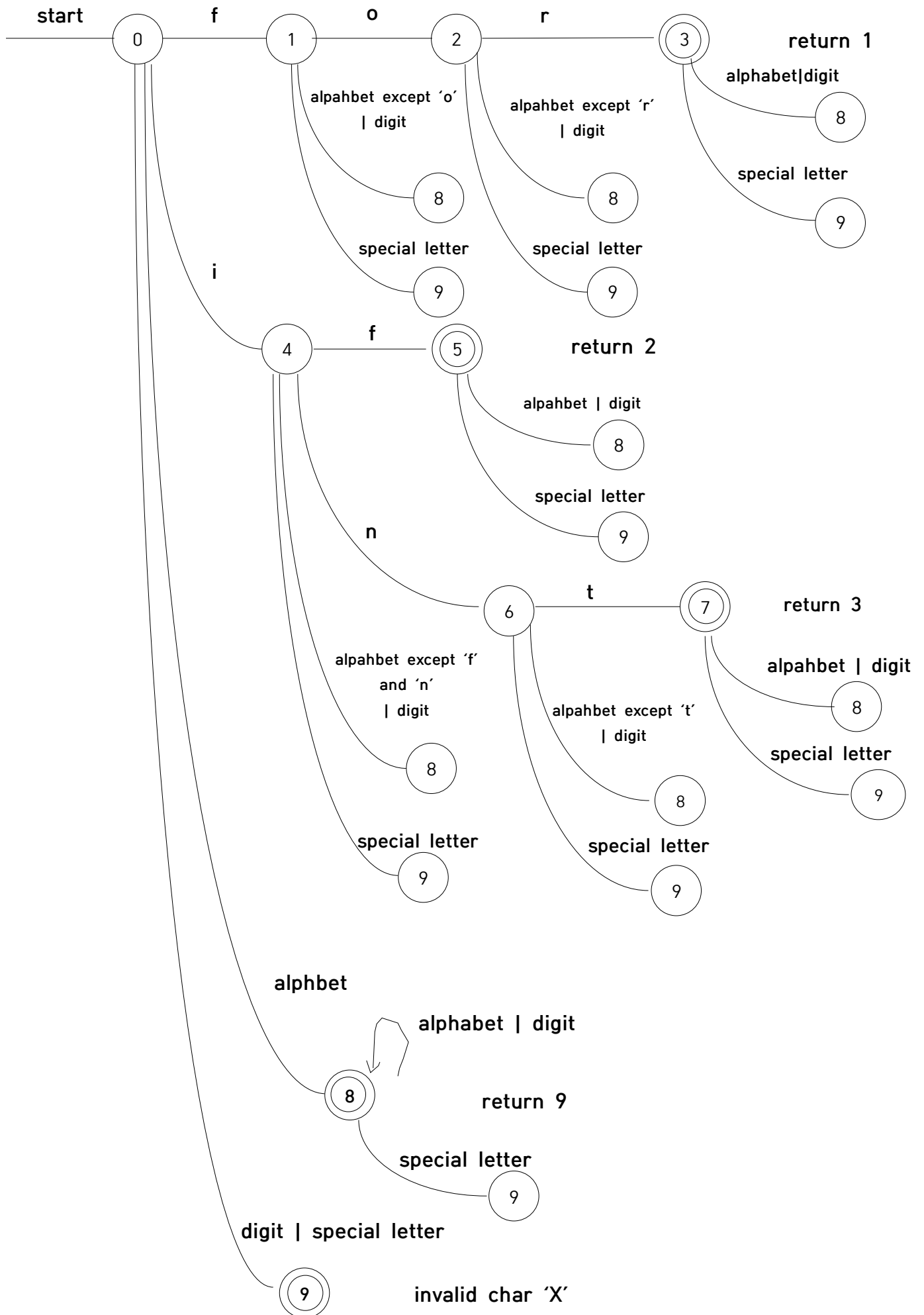
입출력 예

입력	출력	입력	출력
for	1	ifs	9
if	2	if(2 invalid char '('
int	3	B123	9
f1	9	integer	9
i2x	9	int123	9
fff	9	7	invalid char '7'

전이다이어그램은 switch문과 while문을 이용하여 구현한다. switch문의 변수를 state(상태)로 두고, getchar()을 이용해 콘솔창에 작성되어있는 문자열에서 문자들을 차례대로 받아오며 해당하는 연결선을 따라 state의 값을 변경하는 과정을 while문을 이용하여 반복한다. 도달 상태에서 해당하는 숫자 또는 문장을 콘솔창에 출력하도록 한다.

문자열의 끝을 파악하기 위해 입력하는 문자열 뒤에 문자 \$를 추가하도록 해 문자열의 끝을 알리는 역할을 하도록 했다. 즉, 콘솔창 문자열에서 받아온 문자가 \$일 때, 프로그램이 종료된다.

2. 전이 다이어그램 작성



3.코드 설명

```
7  #include<stdio.h>//표준 입출력 함수를 가지고 있는 <stdio.h> 헤더파일
8  #include<stdlib.h>//실행환경 관련 변수들을 가지고 있는 <stdlib.h> 헤더파일
9  #include<ctype.h>//문자 분류에 관한 함수를 가지고 있는 <ctype.h> 헤더파일
10
11  char c;//콘솔창에 입력한 문자열의 문자를 하나씩 받아와 저장하는 변수 lookahead
12  char temp;//state가 9일 때 임시 저장소로 사용하는 변수 temp
13  int state = 0;//전이 다이어그램 위치를 저장하는 변수 state
14
```

7 ~ 9 : 프로그램에 사용된 헤더파일은 다음과 같다.

<stdio.h> : 표준입출력 함수를 가지고 있는 헤더파일

<stdlib.h> : 실행환경 관련 변수들을 가지고 있는 헤더파일

<ctype.h> : 문자 분류에 관한 함수를 가지고 있는 헤더파일

11 ~ 13 : 프로그램에서 사용되는 변수들이다.

```
15  //전이 다이어그램 함수
16  void transaltionDiagram() {
17      while (1) { //while문을 이용하여 반복 수행
18          switch (state) { //변수를 state를 기준으로 switch문으로 state 변경에 대한 처리
19
20              //start(state == 0)
21              case 0: c = getchar(); //문자를 받아와서 변수 c에 저장
22                  //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
23                  if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
24                      state = 0;
25                  //c에 저장된 문자가 알파벳인 경우
26                  else if (isalpha(c) != 0) {
27                      //c에 저장된 문자가 'f'이면 state에 1 저장
28                      if (c == 'f')
29                          state = 1;
30                      //c에 저장된 문자가 'i'이면 state에 4 저장
31                      else if (c == 'i')
32                          state = 4;
33                      //다른 알파벳인 경우 state에 8 저장
34                      else
35                          state = 8;
36                  }
37                  //c에 저장된 문자가 알파벳이 아닌 경우 state에 9 저장
38                  else if (isalpha(c) == 0) {
39                      state = 9;
40                  }
41                  break;

```

16 : 전이 다이어그램 함수를 구현한다

17 : while문을 이용해 반복 수행한다.

18 : 변수 state문에 저장된 값을 기준으로 switch문을 사용한다.

20 ~ 24 : state에 0이 저장되어 있을 경우 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다

25 ~ 36 : c에 저장되어 있는 문자가 알파벳일 경우, f면 1을, i면 4를, 그 외의 알파벳이면 8을 state에 저장한다.

37 ~ 40 : c에 저장되어 있는 문자가 알파벳인 아닐 경우 state에 9를 저장한다.

```

43 //f of 'for'(state == 1)
44 case 1:c = getchar();//문자를 받아와서 변수 c에 저장
45 //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
46 if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
47     state = 1;
48 //c에 저장된 문자가 알파벳인 경우
49 else if (isalpha(c) != 0) {
50     //c에 저장된 문자가 'o'이면 state에 2 저장
51     if (c == 'o')
52         state = 2;
53     //다른 알파벳인 경우 state에 8 저장
54     else
55         state = 8;
56 }//c에 저장된 문자가 숫자인 경우 state에 8 저장
57 else if (isdigit(c) != 0) {
58     state = 8;
59 }
60 else//c에 저장된 문자가 특수문자인 경우 state에 9 저장
61     state = 9;
62 break;
63

```

43 ~ 47 : state에 1이 저장되어 있을 경우 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다

48 ~ 56 : c에 저장되어 있는 문자가 알파벳일 경우, o면 2을, 그 외의 알파벳이면 8을 state에 저장한다.

57 ~ 59 : c에 저장되어 있는 문자가 숫자일 경우, state에 8을 저장한다.

60 ~ 61 : c에 저장되어 있는 문자가 특수문자인 경우 state에 9를 저장한다.

```

64 //o of 'for'(state == 2)
65 case 2:c = getchar();//문자를 받아와서 변수 c에 저장
66 //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
67 if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
68     state = 2;
69 //c에 저장된 문자가 알파벳인 경우
70 else if (isalpha(c) != 0) {
71     //'r'이 저장되어있으면 state에 3 저장
72     if (c == 'r')
73         state = 3;
74     //그 외의 알파벳인 경우 state에 8 저장
75     else
76         state = 8;
77 }//c에 저장된 문자가 숫자인 경우 state에 8 저장
78 else if (isdigit(c) != 0) {
79     state = 8;
80 }
81 else//c에 저장된 문자가 특수문자인 경우 state에 9 저장
82     state = 9;
83 break;
84

```

64 ~ 68 : state에 2이 저장되어 있을 경우 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다.

70 ~ 77 : c에 저장되어 있는 문자가 알파벳일 경우, r이면 3을, 그 외의 알파벳이면 8을 state에 저장한다.

78 ~ 80 : c에 저장되어 있는 문자가 숫자일 경우, state에 8을 저장한다.

81 ~ 82 : c에 저장되어 있는 문자가 특수문자인 경우 state에 9를 저장한다.


```

85 //r of 'for'(state == 3)
86 case 3: c = getchar(); //문자를 받아와서 변수 c에 저장
87 //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
88 if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
89     state = 3;
90 //c에 저장된 '$'인 경우 1을 출력
91 else if (c == '$') {
92     printf("1 ");
93     exit(0);
94 }
95 //c에 저장된 문자가 알파벳이거나 숫자일 경우 state에 8 저장
96 else if (isalpha(c) != 0 || isdigit(c) != 0)
97     state = 8;
98 //c에 저장된 문자가 특수문자인 경우 1 출력하고 state에 9 저장
99 else {
100     printf("1 ");
101     state = 9;
102 }
103 break;
104

```

86 ~ 89 : state에 3이 저장되어 있을 경우 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다

91 ~ 94 : c에 저장된 문자가 \$인 경우 1을 콘솔에 출력하고 종료한다

96 ~ 97 : c에 저장된 문자가 알파벳이거나 숫자일 경우 state에 8을 저장한다

99 ~ 102 : c에 저장된 문자가 \$를 제외한 특수문자인 경우 콘솔에 1을 출력하고 state에 9를 저장한다

```

105 //i of 'if', 'int'(state == 4)
106 case 4: c = getchar(); //문자를 받아와서 변수 c에 저장
107 //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
108 if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
109     state = 4;
110 //c에 저장된 문자가 알파벳일 경우
111 else if (isalpha(c) != 0) {
112     //c에 저장된 문자가 'f'인 경우 state에 5 저장
113     if (c == 'f')
114         state = 5;
115     //c에 저장된 문자가 'n'인 경우 state에 6 저장
116     else if (c == 'n')
117         state = 6;
118     //다른 알파벳인 경우 state에 8 저장
119     else
120         state = 8;
121 } //c에 저장된 문자가 숫자인 경우 state에 8 저장
122 else if (isdigit(c) != 0) {
123     state = 8;
124 }
125 else //c에 저장된 문자가 특수문자인 경우 state에 9 저장
126     state = 9;
127 break;
128

```

106 ~ 109 : state에 4가 저장되어 있을 경우 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다.

111 ~ 121 : c에 저장되어 있는 문자가 알파벳일 경우, f면 5를, n이면 6을, 그 외의 알파벳이면 8을 state에 저장한다.

122 ~ 124 : c에 저장되어 있는 문자가 숫자일 경우, state에 8을 저장한다.

125 ~ 126 : c에 저장되어 있는 문자가 특수문자인 경우 state에 9를 저장한다.

```

129 //f of 'if'(state == 5)
130 case 5: c = getchar(); //문자를 받아와서 변수 c에 저장
131 //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
132 if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
133     state = 5;
134 //c에 저장된 문자가 '$'일 경우 콘솔창에 2 출력
135 else if (c == '$') {
136     printf("2 ");
137     exit(0);
138 }
139 //c에 저장된 문자가 알파벳이거나 숫자일 경우 state에 8 저장
140 else if (isalpha(c) != 0 || isdigit(c) != 0)
141     state = 8;
142 //c에 저장된 문자가 특수문자일 경우 2를 출력하고 state에 9 저장
143 else {
144     printf("2 ");
145     state = 9;
146 }
147 break;
148

```

130 ~ 133 : state에 5가 저장되어 있을 경우 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다

135 ~ 138 : c에 저장된 문자가 \$인 경우 2를 콘솔에 출력하고 종료한다

140 ~ 141 : c에 저장된 문자가 알파벳이거나 숫자일 경우 state에 8을 저장한다

143 ~ 146 : c에 저장된 문자가 \$를 제외한 특수문자인 경우 콘솔에 2를 출력하고 state에 9를 저장한다

```

149 //n of 'int'(state == 6)
150 case 6: c = getchar(); //문자를 받아와서 변수 c에 저장
151 //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
152 if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
153     state = 6;
154 //c에 저장된 문자가 알파벳인 경우
155 else if (isalpha(c) != 0) {
156     //c에 저장된 문자가 't'에 저장된 경우 state에 7 저장
157     if (c == 't')
158         state = 7;
159     //다른 알파벳인 경우 state에 8 저장
160     else
161         state = 8;
162 } //c에 저장된 문자가 숫자인 경우 state에 8 저장
163 else if (isdigit(c) != 0) {
164     state = 8;
165 }
166 else //c에 저장된 문자가 특수문자인 경우 state에 9 저장
167     state = 9;
168 break;
169

```

150 ~ 153 : state에 6이 저장되어 있을 경우 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다.

155 ~ 162 : c에 저장되어 있는 문자가 알파벳일 경우, t면 7을, 그 외의 알파벳이면 8을 state에 저장한다.

163 ~ 165 : c에 저장되어 있는 문자가 숫자일 경우, state에 8을 저장한다.

166 ~ 167 : c에 저장되어 있는 문자가 특수문자인 경우 state에 9를 저장한다.


```

170 //t of 'int'(state == 7)
171 case 7: c = getchar(); //문자를 받아와서 변수 c에 저장
172 //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
173 if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
174     state = 7;
175 //c에 저장된 문자가 '$'일 경우 콘솔창에 3 출력하고 종료
176 else if (c == '$') {
177     printf("3 ");
178     exit(0);
179 }
180 //c에 저장된 문자가 알파벳이거나 숫자일 경우 state에 8 저장
181 else if (isalpha(c) != 0 || isdigit(c) != 0)
182     state = 8;
183 //c에 저장된 문자가 특수문자일 경우 콘솔창에 3 출력하고 state에 9 저장
184 else {
185     printf("3 ");
186     state = 9;
187 }
188 break;
189

```

171 ~ 174: state에 7이 저장되어 있을 경우 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다

176 ~ 179 : c에 저장된 문자가 \$인 경우 3을 콘솔에 출력하고 종료한다

96 ~ 97 : c에 저장된 문자가 알파벳이거나 숫자일 경우 state에 8을 저장한다

99 ~ 102 : c에 저장된 문자가 \$를 제외한 특수문자인 경우 콘솔에 3을 출력하고 state에 9를 저장한다

```

190 //alphabet | digit(state == 8)
191 case 8:
192     //c에 저장된 문자가 알파벳이거나 숫자인 경우
193     if (isalpha(c) != 0 || isdigit(c) != 0) {
194         c = getchar(); //c에 문자를 받아와 저장
195         //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
196         if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
197             state = 8;
198         //state에 8을 저장한다
199         else {
200             state = 8;
201         }
202     }
203     //c에 저장된 문자가 특수문자일 경우
204     else {
205         //c에 저장된 문자가 '$'인 경우 콘솔창에 9를 출력하고 종료한다.
206         if (c == '$') {
207             printf("9 ");
208             exit(0);
209         }
210         // '$'가 아닌 경우 콘솔창에 9를 출력하고 state에 9를 저장한다.
211         else {
212             printf("9 ");
213             state = 9;
214         }
215     }
216     break;
217

```

191 ~ 202 : state에 8이 저장되어 있을 경우, c에 저장되어 있는 문자가 알파벳이거나 숫자면 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다.

204 ~ 215 : c에 저장되어 있는 문자가 특수문자일 경우, \$가 저장되어 있으면 콘솔에 9를 출력하고 종료하고, 다른 특수문자가 들어있을 경우 9를 출력하고 state에 9를 저장한다.

```

218 //digit | special(state == 9)
219 case 9:
220     temp = c; //c에 저장되어 있는 문자를 temp에 저장한다
221     c = getchar(); //문자를 받아와서 변수 c에 저장
222     //c에 저장된 문자가 blank, tab, newline, NUL일 경우에는 돌아가서 다시 받아온다
223     if (c == ' ' || c == '\t' || c == '\n' || c == '\0')
224         state = 9;
225     //c에 저장된 문자가 '$'인 경우 temp에 저장된 문자를 콘솔창에 출력하고 종료한다.
226     else if (c == '$') {
227         printf("invalid char '%c' ", temp);
228         exit(0);
229     }
230     //c에 저장된 문자가 알파벳이거나 숫자면 temp에 저장된 문자를 콘솔창에 출력하고 state에 8을 저장한다.
231     else if (isalpha(c) != 0 || isdigit(c) != 0) {
232         printf("invalid char '%c' ", temp);
233         state = 8;
234     }
235     //c에 저장된 문자가 특수문자면 temp에 저장된 문자를 콘솔창에 출력하고 state에 9를 저장한다.
236     else {
237         printf("invalid char '%c' ", temp);
238         state = 9;
239     }
240     break;
241 }
242 }
243 }
244 //메인함수
245 void main() {
246     transaitonDiagram(); //전이 다이어그램 함수 실행
247 }

```

219 ~ 224 : state에 9가 저장되어 있는 경우, c에 들어있는 문자를 temp에 저장한 뒤, 콘솔창에서 다음문자를 받아와 c에 저장한다. 저장된 문자가 blank, tab, newline, NUL일 경우에는 되돌아가서 다시 받아온다.

226 ~ 229 : c에 저장된 문자가 \$인 경우 'invalid char X'을 콘솔에 출력하고 종료한다

231 ~ 234 : c에 저장된 문자가 알파벳이거나 숫자면 temp에 저장되어 있는 문자를 이용해 'invalid char X'을 콘솔에 출력하고, state에 8을 저장한다.

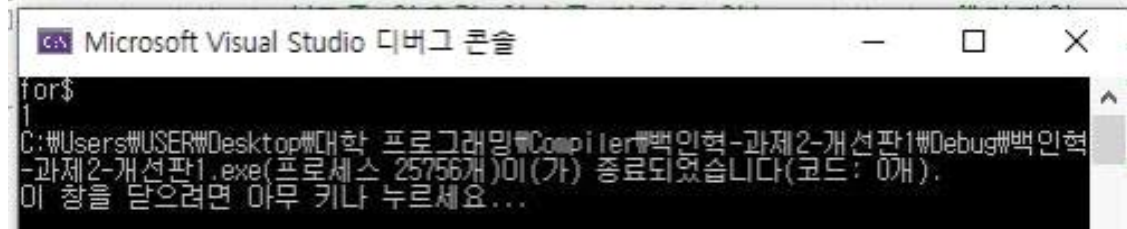
236 ~ 239 : c에 저장된 문자가 \$를 제외한 특수문자면 temp에 저장된 문자를 이용해 문자를 이용해 'invalid char X'을 콘솔에 출력하고, state에 9를 저장한다.

245 ~ 247 : 메인함수에서 전이 다이어그램 함수 transaitonDiagram()을 수행한다.

4.실행 예시

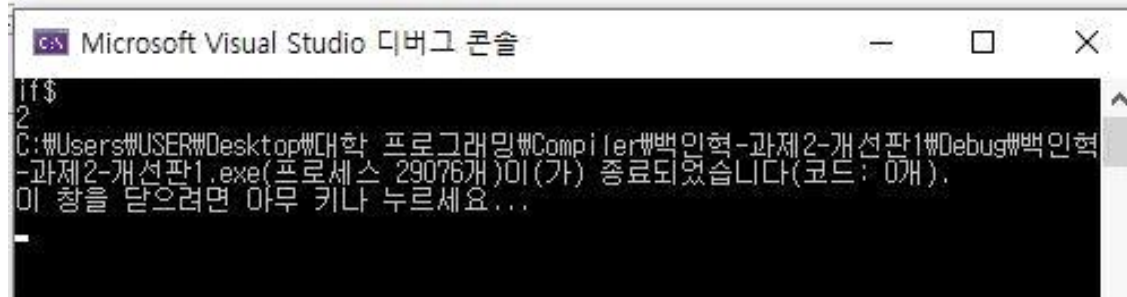
예시1) for

```
/*프로그램명 : 2022-1 컴파일러 과제 2
* 작성자 : 2020039096 백인혁
* 최초 작성 완료일 : 2022.04.09
* 1차 수정 : 2022.04.11
* 특이사항 : 문자열 끝에 $가 입력하여 문자열이 끝났음을 알린다. ex) for
*/
```



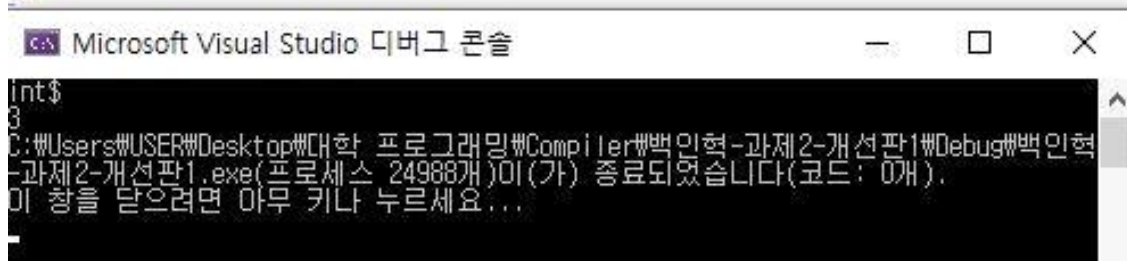
예시2) if

```
/*프로그램명 : 2022-1 컴파일러 과제 2
* 작성자 : 2020039096 백인혁
* 최초 작성 완료일 : 2022.04.09
* 1차 수정 : 2022.04.11
* 특이사항 : 문자열 끝에 $가 입력하여 문자열이 끝났음을 알린다. ex) for
*/
```



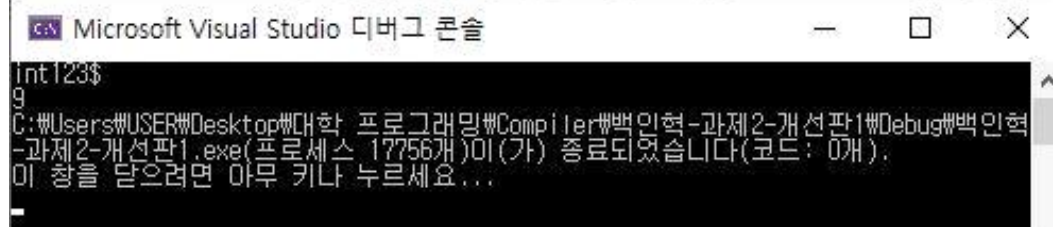
예시3) int

```
/*프로그램명 : 2022-1 컴파일러 과제 2
* 작성자 : 2020039096 백인혁
* 최초 작성 완료일 : 2022.04.09
* 1차 수정 : 2022.04.11
* 특이사항 : 문자열 끝에 $가 입력하여 문자열이 끝났음을 알린다. ex) fo
*/
```



예시4) int123

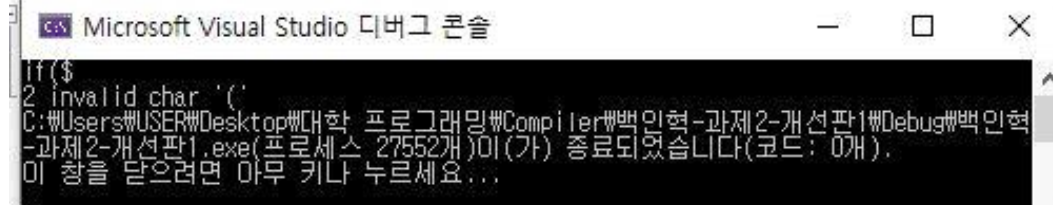
```
3 /*프로그램명 : 2022-1 컴파일러 과제 2
 * 작성자 : 2020039096 백인혁
 * 최초 작성 완료일 : 2022.04.09
 * 1차 수정 : 2022.04.11
 * 특이사항 : 문자열 끝에 $가 입력하여 문자열이 끝났음을 알린다. ex) fo
 */
```



```
Microsoft Visual Studio 디버그 콘솔
int123$
9
C:\Users\USER\Desktop\대학 프로그래밍\Compiler\백인혁-과제2-개선판1\Debug\백인혁-과제2-개선판1.exe(프로세스 17756개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

예시5) if(

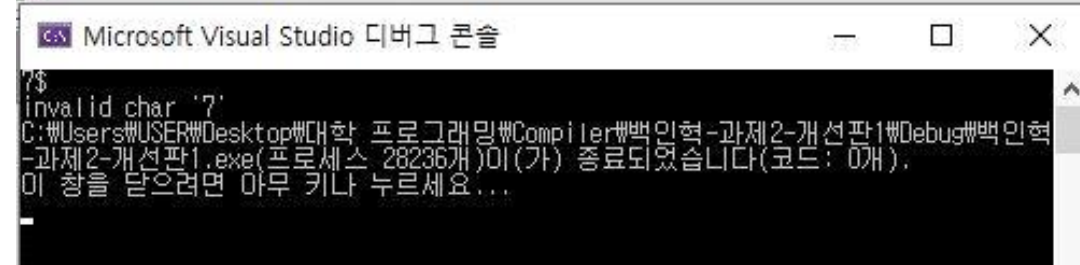
```
3 /*프로그램명 : 2022-1 컴파일러 과제 2
 * 작성자 : 2020039096 백인혁
 * 최초 작성 완료일 : 2022.04.09
 * 1차 수정 : 2022.04.11
 * 특이사항 : 문자열 끝에 $가 입력하여 문자열이 끝났음을 알린다. ex) fo
 */
```



```
Microsoft Visual Studio 디버그 콘솔
if($
2 invalid char '('
C:\Users\USER\Desktop\대학 프로그래밍\Compiler\백인혁-과제2-개선판1\Debug\백인혁-과제2-개선판1.exe(프로세스 27552개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

예시6) 7

```
3 /*프로그램명 : 2022-1 컴파일러 과제 2
 * 작성자 : 2020039096 백인혁
 * 최초 작성 완료일 : 2022.04.09
 * 1차 수정 : 2022.04.11
 * 특이사항 : 문자열 끝에 $가 입력하여 문자열이 끝났음을 알린다. ex) fo
 */
```



```
Microsoft Visual Studio 디버그 콘솔
7$
invalid char '7'
C:\Users\USER\Desktop\대학 프로그래밍\Compiler\백인혁-과제2-개선판1\Debug\백인혁-과제2-개선판1.exe(프로세스 28236개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```