# Practical-10

# Write a C program to parse a given string using Predictive parsing for given grammar. type → simple | ↑id | array [ simple ] of type simple → integer | char | num dotdot num

#include<stdio.h>

#include<conio.h>

#include<string.h>

voidnext\_token();

void match(char t[]);

void terminals();

void simple();

void ERROR();

charlookahead[30]={'\0'};

FILE \*fp1;

charstr[100]={'\0'};

inti,flag=0;

void main()

{

char c;

clrscr();

fp1 = fopen("D:/grammar.txt","r");

printf("The Grammar is : \n");

while((c=getc(fp1)) != EOF)

{

printf("%c",c);

}

printf("\n\nEnter any string : ");

gets(str);

i=0;

next\_token();

terminals();

printf("\n\nCongratulations ! String is valid");

getch();

}

voidnext\_token()

{

intj,k;

for(k=0;k<30;k++)

lookahead[k] = '\0';

for(j=0;str[i] != ' ' &&str[i] != '\0';j++)

{

lookahead[j] = str[i];

i++;

}

i++;

lookahead[j] = '\0';

}

void terminals()

{

if(strcmp(lookahead,"int")== 0 ||

strcmp(lookahead,"char")== 0 || strcmp(lookahead,"num")==0)

{

simple();

if(strlen(lookahead) > 0 )

ERROR();

}

else if( strcmp(lookahead,"^") == 0)

{

match("^");

match("id");

if(strlen(lookahead) > 0 )

ERROR();

}

else if( strcmp(lookahead,"array") == 0)

{

match("array");

match("[");

simple();

match("]");

match("of");

terminals();

if(strlen(lookahead) > 0 )

ERROR();

}

else

ERROR();

}

void match(char t[])

{

if(strcmp(lookahead,t) == 0)

{

next\_token();

}

else

{

ERROR();

}

}

void simple()

{

if(strcmp(lookahead,"int") == 0)

{

match("int");

}

else if( strcmp(lookahead,"char") == 0)

{

match("char");

}

else if( strcmp(lookahead,"num") == 0)

{

match("num");

match(".");

match(".");

match("num");

}

else

ERROR();

}

void ERROR()

{

printf("\n\nError : Invalid String ");

getch();

exit(0);

}

OUTPUT:

Enter any string : array [char] of int

Congratulations ! String is valid