```
#include <cstdio>
#include <cstring>
#include <iostream>
using namespace std;
int stack_position = 0,
        stack_ptr = 0,
        in_str_ptr = 0,
        input_len = 0;

char in_str[16], //input string
stk[15];    //stack

void check(){


    // begin with checking stack_ptr first.
    for(stack_position = 0; stack_position < input_len; stack_position++){

        if(stk[stack_position] == 'i'){
            printf("REDUCE TO F -> i");
            stk[stack_position] = 'F';
            stk[stack_position + 1] = '\0';

            //printing stack info
            printf("\n$%s\t\t%s$\t", stk, in_str);

            if(stk[0] == 'T'){
                if(stk[1] == '*') {
                    if (stk[2] == 'F') {
                        printf("REDUCE TO T -> T * F");

                        stk[0] = 'T';
                        stk[1] = '\0';

                        //printing stack info
                        printf("\n$%s\t\t%s$\t", stk, in_str);
                        stack_ptr=0;
                        break;
                    }
                }
            }
        }

        if(stk[stack_position] == 'F') {
            printf("REDUCE TO T -> F");
            stk[stack_position] = 'T';
            stk[stack_position + 1] = '\0';

            //printing stack info
            printf("\n$%s\t\t%s$\t", stk, in_str);

            if (stk[0] == 'T') {
                if (stk[1] == '+') {
                    if (stk[2] == 'T') {
                        printf("REDUCE TO E -> T + T");

                        stk[0] = 'E';
                        stk[1] = '\0';
                        //printing stack info
                        printf("\n$%s\t\t%s$\t", stk, in_str);
                        stack_ptr = 0;
                        break;
                    }
                }


            }
        }
    }


}

int main(){
    printf("GRAMMAR is -\nE  -> T + T | T \nT ->  T * F | F \nF ->  i\n");

    strcpy(in_str,"i*i+i");

    printf("Input string is: %s\n", in_str);
    // strlen(in_str) will return the length of in_str
    input_len =strlen(in_str);
    // This will print Lables (column name)
    printf("\nStack    Input     Action");

    // This will print the initial values of stack and input
    printf("\n$\t\t%s$\t", in_str);   //stack = $

    // This will run up to length of input string
    for(stack_ptr = 0; in_str_ptr < input_len; stack_ptr++, in_str_ptr++){
        printf("SHIFT");   //start with SHIFT
        // Shift- pushing into stack
        stk[stack_ptr] = in_str[in_str_ptr];
        stk[stack_ptr + 1] = '\0';  //'0' is end of string

        // Remove char from in_str - make it blank
        in_str[in_str_ptr]=' ';

        // Printing action
        printf("\n$%s\t\t%s$\t", stk, in_str);
        //check the stack to see if it contains any production rules


        check();
    }

    // if top of the stack is E(starting symbol, then it will accept the input
    if(stk[0] == 'E' && stk[1] == '\0')
        printf("Accept\n");
    else //else reject
        printf("Reject\n");
}
```