EXPERIMENT - 7

SHIFT REDUCE PARSING

AIM: To implement a program for shift Reduce parsing

Algorithm:

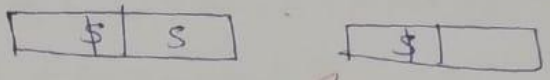
Shift Reduce parser requires two data Structures
→ Input Buffer → stack

Steps: ① Insert $ at bottom of stack & right end of input string in input Buffer.

* Shift parser shifts zero or more input symbols onto the stack until the handle is on top of stack.

* Reduce parser replace the handle on top of stack to left side of production.

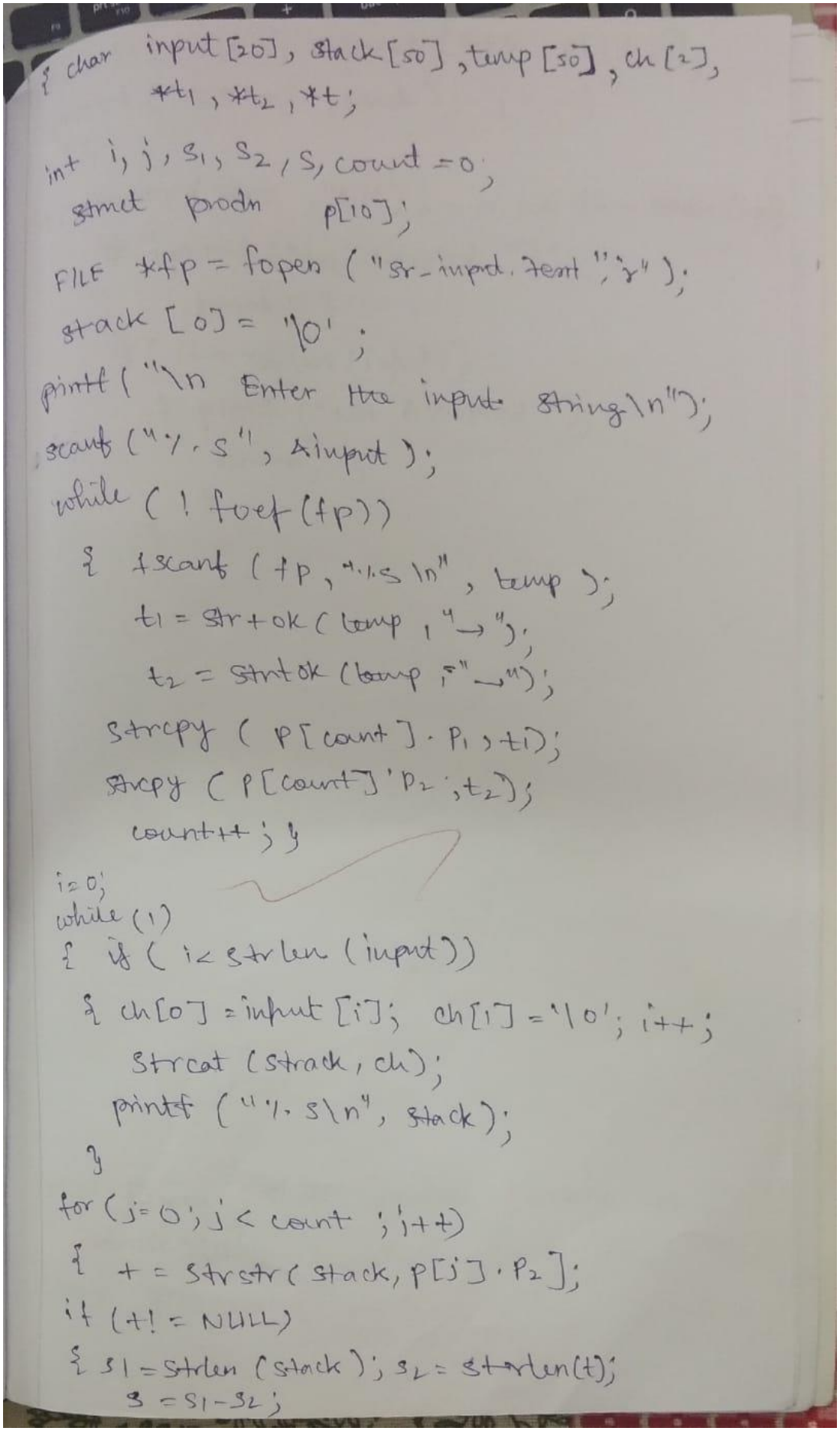
* Except step 3 & step 4 will be repeated until it has detected an error or until the stack includes start symbol(s) and input Buffer is empty i.e it contain $.



PROGRAM:

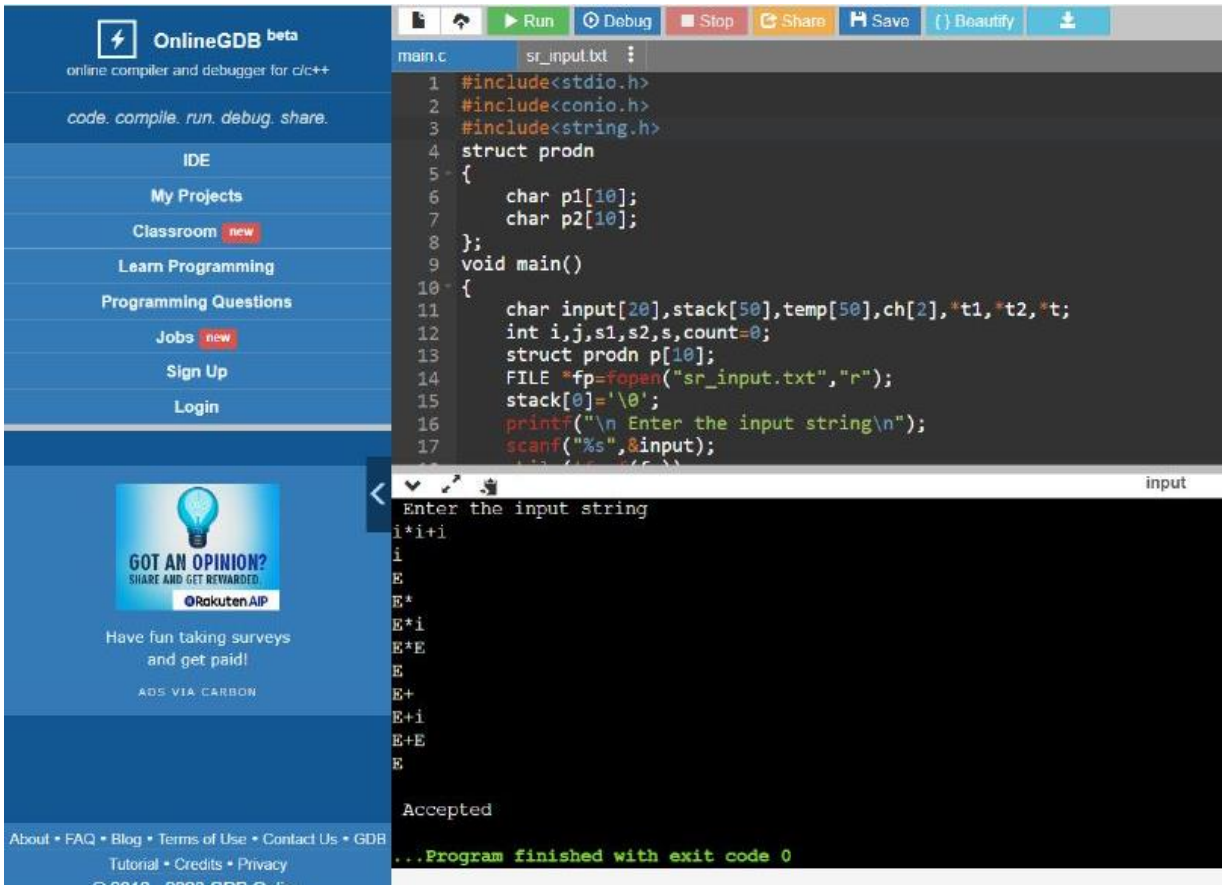
```
#include <stdio.h>
#include <conio.h>
#include <string.h>
struct prodn
{ char P1[10];
  char P2[10]; };
void main()
```

DURGA CHANDANA SREE M

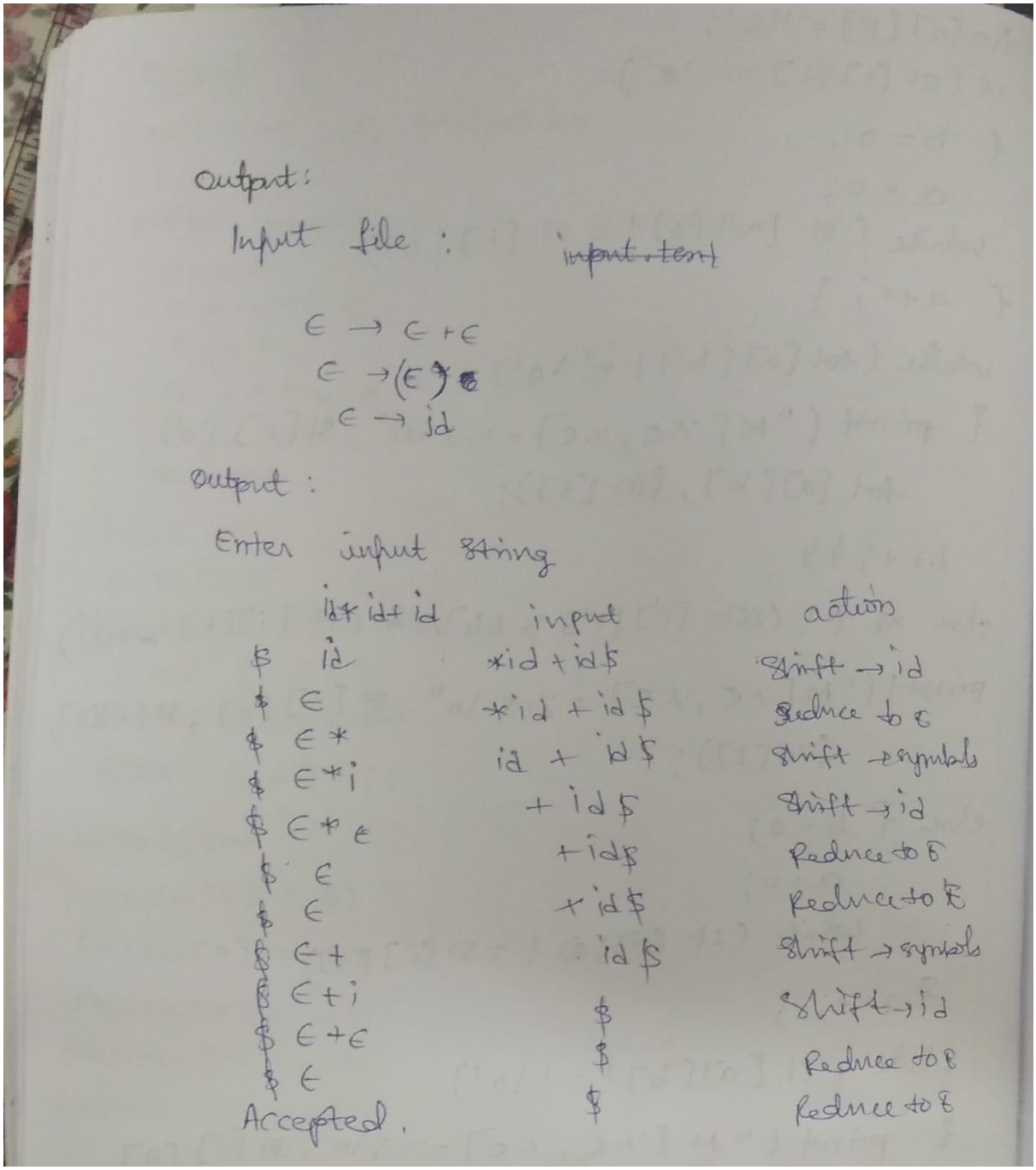


```
{ char input[20], stack[50], temp[50], ch[2],
      *t1, *t2, *t;
int i, j, s1, s2, s, count=0;
 struct prodn  p[10];
FILE *fp = fopen ("sr-input.text","r");
 stack[0] = '0';
printf("\n Enter the input string\n");
scanf("%s", &input);
while (! feof(fp))
  { fscanf (fp, "%s\n", temp);
     t1 = strtok(temp, "->");
     t2 = strtok(temp, "->");
   strcpy ( p[count].p1, t1);
   strcpy (p[count].p2, t2);
    count++; }
i=0;
while(1)
{ if ( i< strlen (input))
  { ch[0] = input[i]; ch[1] = '\0'; i++;
    strcat (stack, ch);
   printf ("%s\n", stack);
  }
for(j=0; j< count ;j++)
{  t = strstr( stack, p[j].p2);
if (t != NULL)
{ s1 = strlen (stack); s2 = strlen(t);
   s = s1-s2;
```

DURGA CHANDANA SREE M



DURGA CHANDANA SREE M



Output:

Input file : input.text

$E \rightarrow E+E$

$E \rightarrow (E)$

$E \rightarrow id$

Output:

Enter input string

id*id+id

$	id	*id+id$	shift → id
$	E	*id+id$	Reduce to E
$	E*	id+id$	shift → symbols
$	E*i	+id$	shift → id
$	E*E	+id$	Reduce to E
$	E	+id$	Reduce to E
$	E	+id$	shift → symbols
$	E+	id$	shift → id
$	E+i	$	Reduce to E
$	E+E	$	Reduce to E
$	E	$	

Accepted.

```
stack[s] = '\0'; strcat(stack, p[j].p);
printf("%s\n", stack); j = -1;
}}
if(strcmp(stack, "E") == 0 && i == strlen(input))
    {printf("\n Accepted ");
     break; }
if(i == strlen(input))
   { printf("NOT Accepted ");
     break; }
  }
}
```

RESULT: Hence shift reduce parsing is successfully executed with the above program.