PROGRAM

```
// Shift Reduce Parser for the following grammar:
// (1) E' -> E
// (2) E -> aEa
// (3) E -> b

// The Parsing table is as follows:
//     a   b   $   |   E
// 0   s2 s3      |   1
// 1            ac  |
// 2   s2 s3      |   4
// 3   r3      r3  |
// 4   s5          |
// 5   r2      r2  |

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

char input[50], stk[50], a[50];
int len, top = 0, flag = 1;

void goto_func()
{
   if(stk[top]=='E' && stk[top-1] == '0')
      stk[++top] = '1';
   else if(stk[top]=='E' && stk[top-1]=='2')
      stk[++top] = '4';
   else
   {
      flag = 0;
      printf("\nError");
      exit(0);
   }
}
void take_action(int idx)
{
   printf("\n%s\t\t%s\t\t",stk,a);
   if(stk[top]=='1' && a[idx]=='$')
   {
      printf("ACCEPT\n");
      flag = 1;
```

```
    return;
}
else if(stk[top]=='0' && a[idx]=='a') //s2
{
    stk[++top] = a[idx];
    a[idx] = ' ';
    stk[++top] = '2';
    printf("SHIFT\n");
}
else if(stk[top]=='0' && a[idx]=='b') //s3
{
    stk[++top] = a[idx];
    a[idx] = ' ';
    stk[++top] = '3';
    printf("SHIFT\n");
}
else if(stk[top]=='2' && a[idx]=='a') //s2
{
    stk[++top] = a[idx];
    a[idx] = ' ';
    stk[++top] = '2';
    printf("SHIFT\n");
}
else if(stk[top]=='2' && a[idx]=='b') //s3
{
    stk[++top] = a[idx];
    a[idx] = ' ';
    stk[++top] = '3';
    printf("SHIFT\n");
}
else if(stk[top]=='4' && a[idx]=='a') //s5
{
    stk[++top] = a[idx];
    a[idx] = ' ';
    stk[++top] = '5';
    printf("SHIFT\n");
}
else if(stk[top]=='3' && a[idx]=='a') //r3
{
    stk[top] = ' ';
    top--;
    stk[top] = 'E';
    printf("REDUCE by E -> b\n");
```

```
        goto_func();
        take_action(idx);
    }
    else if(stk[top]=='3' && a[idx]=='$') //r3
    {
        stk[top] = ' ';
        top--;
        stk[top] = 'E';
        printf("REDUCE by E -> b\n");
        goto_func();
        take_action(idx);
    }
    else if(stk[top]=='5' && a[idx]=='$') //r3
    {
        for(int i=0;i<5;i++)
        {
            stk[top] = ' ';
            top--;
        }
        stk[top] = 'E';
        printf("REDUCE by E -> aEa\n");
        goto_func();
        take_action(idx);
    }
    else if(stk[top]=='5' && a[idx]=='a') //r2
    {
        for(int i=0;i<5;i++)
        {
            stk[top] = ' ';
            top--;
        }
        stk[top] = 'E';
        printf("REDUCE by E -> aEa\n");
        goto_func();
        take_action(idx);
    }
    else
    {
        flag = 0;
        return;
    }
}
```

```
void main()
{
   printf("\nGrammar is:\nE -> aEa\nE -> b\n");
   printf("\nEnter The String: ");
   scanf("%s",input);

   strcpy(a,strcat(input,"$"));
   stk[top]='0';
   len = strlen(a);
   printf("\nStack\t\tInput\t\tAction");
   for(int i=0;i<len;i++)
   {
      take_action(i);
      if(flag==0)
         break;
   }
   if(flag == 1)
      printf("\nString Accepted");
   else
      printf("\nERROR!\nString Rejected");
}
```

OUTPUT

```
E:\Semester 7\Compiler Design Lab\Practice>gcc shift2.c

E:\Semester 7\Compiler Design Lab\Practice>a

Grammar is:
E -> aEa
E -> b

Enter The String: aabaa

Stack           Input           Action
0               aabaa$          SHIFT

0a2              abaa$          SHIFT

0a2a2             baa$          SHIFT

0a2a2b3            aa$          REDUCE by E -> b

0a2a2E4            aa$          SHIFT

0a2a2E4a5                   a$          REDUCE by E -> aEa

0a2E4                       a$          SHIFT

0a2E4a5                      $          REDUCE by E -> aEa

0E1                          $          ACCEPT

String Accepted
E:\Semester 7\Compiler Design Lab\Practice>
```