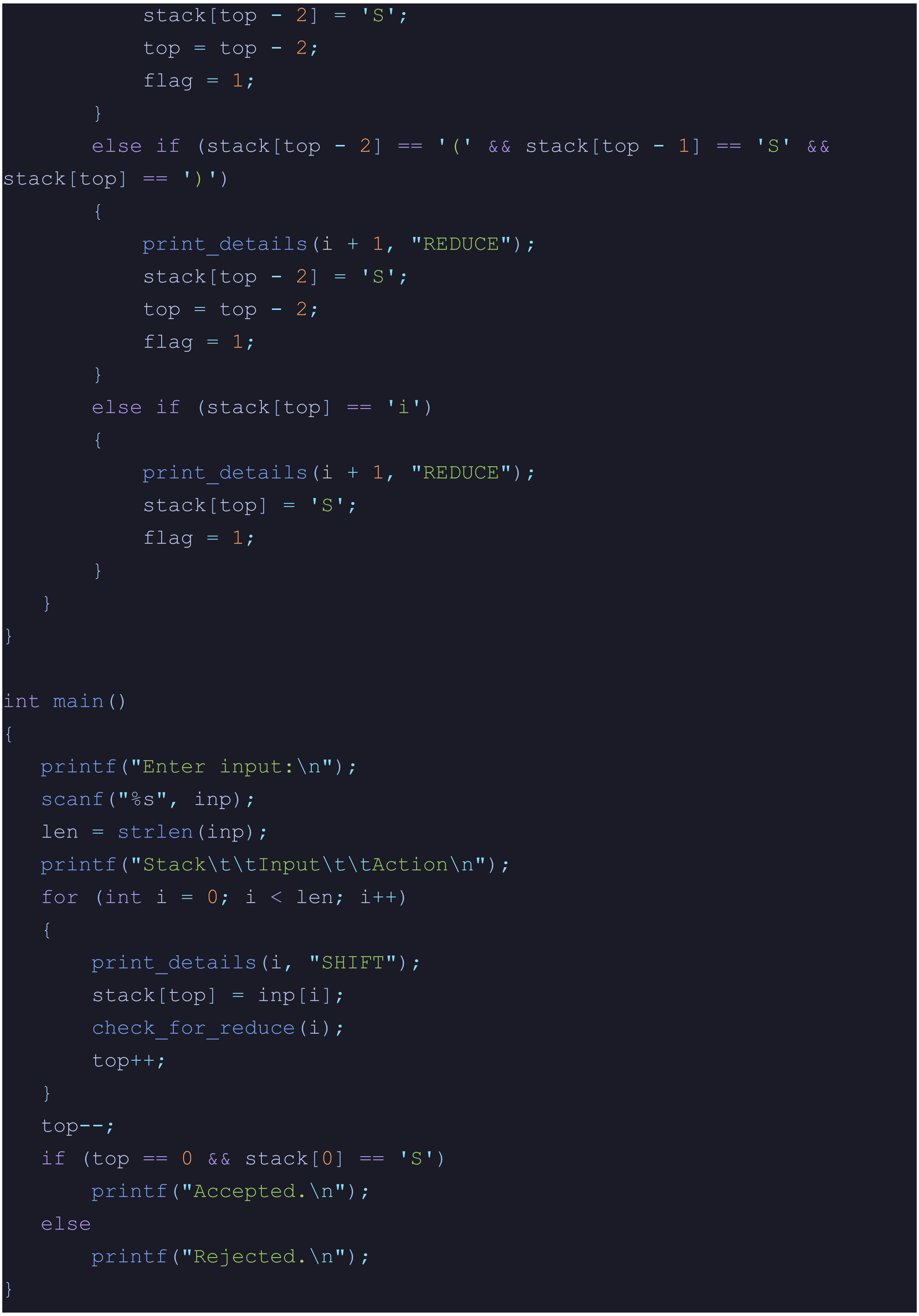
Name: Gokulnath M Prabhu
Class: CS7B
Roll No: 21

Lab Cycle 3 - Experiment 16

Construct a Shift Reduce Parser for a given language.

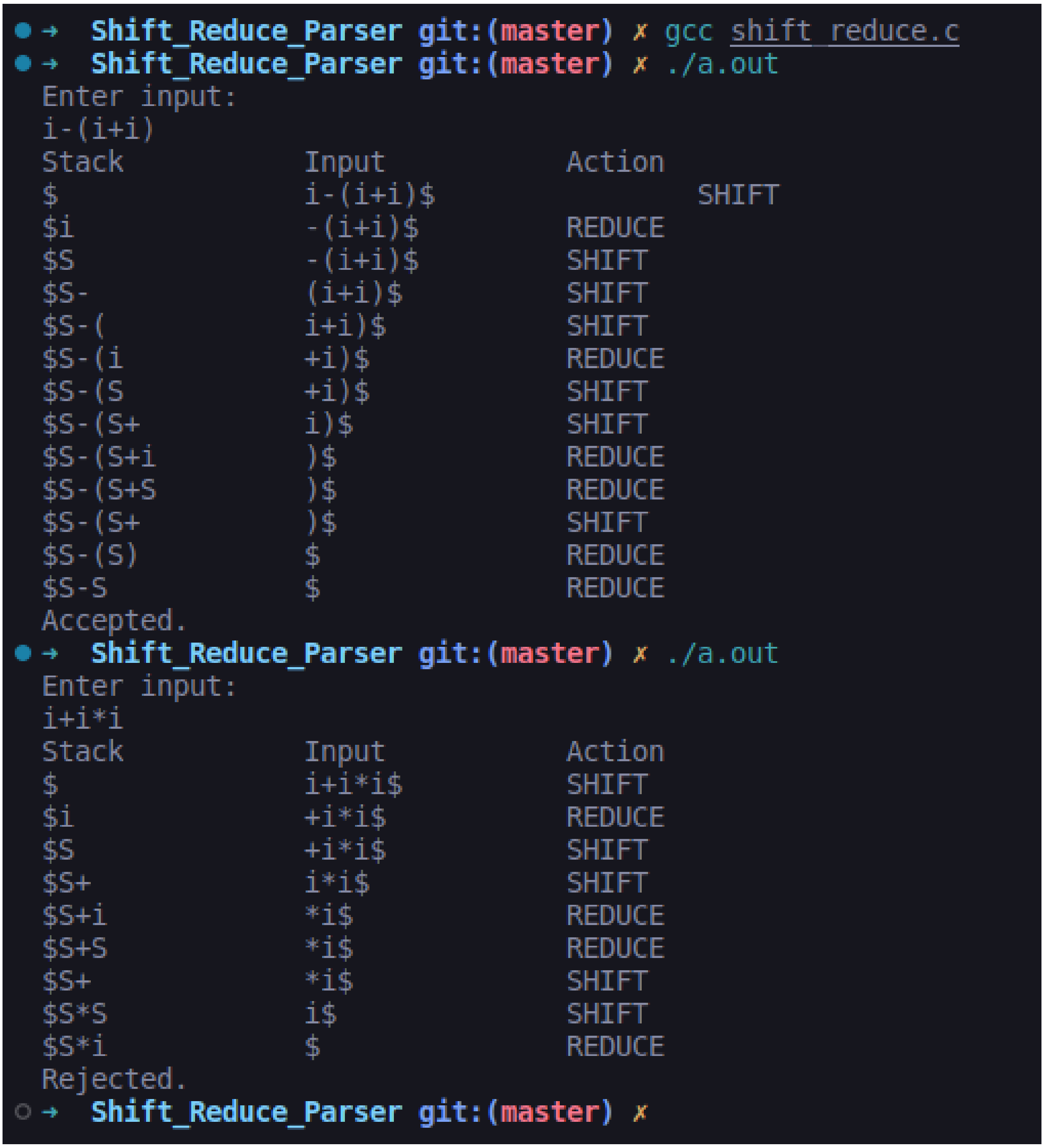
Code:

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

char inp[100];
int len;
char stack[100];
int top = 0;

void print_details(int ind, char *action)
{
   printf("$");
   for (int i = 0; i <= top; i++)
       printf("%c", stack[i]);
   printf("\t\t");
   for (int i = ind; i < len; i++)
       printf("%c", inp[i]);
   printf("$\t\t%s\n", action);
}

void check_for_reduce(int i)
{
   int flag = 1;
   while (flag)
   {
       flag = 0;
       if (stack[top - 2] == 'S' && stack[top - 1] == '+' && stack[top]
== 'S')
       {
           print_details(i + 1, "REDUCE");
           stack[top - 2] = 'S';
           top = top - 2;
           flag = 1;
       }
       else if (stack[top - 2] == 'S' && stack[top - 1] == '-' &&
stack[top] == 'S')
       {
           print_details(i + 1, "REDUCE");
```



```c
            stack[top - 2] = 'S';
            top = top - 2;
            flag = 1;
        }
        else if (stack[top - 2] == '(' && stack[top - 1] == 'S' &&
stack[top] == ')')
        {
            print_details(i + 1, "REDUCE");
            stack[top - 2] = 'S';
            top = top - 2;
            flag = 1;
        }
        else if (stack[top] == 'i')
        {
            print_details(i + 1, "REDUCE");
            stack[top] = 'S';
            flag = 1;
        }
    }
}

int main()
{
    printf("Enter input:\n");
    scanf("%s", inp);
    len = strlen(inp);
    printf("Stack\t\tInput\t\tAction\n");
    for (int i = 0; i < len; i++)
    {
        print_details(i, "SHIFT");
        stack[top] = inp[i];
        check_for_reduce(i);
        top++;
    }
    top--;
    if (top == 0 && stack[0] == 'S')
        printf("Accepted.\n");
    else
        printf("Rejected.\n");
}
```

Output:



```
● ➜  Shift_Reduce_Parser git:(master) ✗ gcc shift_reduce.c
● ➜  Shift_Reduce_Parser git:(master) ✗ ./a.out
  Enter input:
  i-(i+i)
  Stack            Input            Action
  $                i-(i+i)$                 SHIFT
  $i               -(i+i)$          REDUCE
  $S               -(i+i)$          SHIFT
  $S-              (i+i)$           SHIFT
  $S-(             i+i)$            SHIFT
  $S-(i            +i)$             REDUCE
  $S-(S            +i)$             SHIFT
  $S-(S+           i)$              SHIFT
  $S-(S+i          )$               REDUCE
  $S-(S+S          )$               REDUCE
  $S-(S+           )$               SHIFT
  $S-(S)           $                REDUCE
  $S-S             $                REDUCE
  Accepted.
● ➜  Shift_Reduce_Parser git:(master) ✗ ./a.out
  Enter input:
  i+i*i
  Stack            Input            Action
  $                i+i*i$           SHIFT
  $i               +i*i$            REDUCE
  $S               +i*i$            SHIFT
  $S+              i*i$             SHIFT
  $S+i             *i$              REDUCE
  $S+S             *i$              REDUCE
  $S+              *i$              SHIFT
  $S*S             i$               SHIFT
  $S*i             $                REDUCE
  Rejected.
○ ➜  Shift_Reduce_Parser git:(master) ✗
```