Lab-5 INTRODUCTION TO FLEX

- Name - Aditya Sinha
- Reg.No - 230905218
- CSE-A-27

1. Grammar (Q1)

- Original: S → a | > | (T); T → T , S | S (left recursion in T)
- LL(1) elimination applied for T:
  - T → S T'
  - T' → , S T' | ε

Code (q1.c)

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

void S();
void T();
void Tp();
void invalid();
void valid();


int curr = 0;
char str[100];

void main() {
    printf("Enter  String:  ");
    scanf("%s", str);
    S();
    if(str[curr] == '$') valid();
    else {
        printf("%c\n", str[curr]);
        invalid();
    }
}
void S() {
    if(str[curr] == 'a' || str[curr] == '>') {
        curr++;
        return;
    }
    if(str[curr] == '(') {
        curr++;
        T();
        if(str[curr] == ')') {
            curr++;
            return;
        }
```

```
        invalid();
    }
    invalid();
}

void T() {
    if(str[curr] == 'a' || str[curr] == '>' || str[curr] == '(') {
        S();
        Tp();
        return;
    }
    invalid();
}

void Tp() {
    if(str[curr] == ',') {
        curr++;
        S();
        Tp();
        return;
    }
    if(str[curr] == ')') return;
    invalid();
}

void invalid() {
   printf("-----------------ERROR!----------------\n");
   exit(0);
}

void valid() {
  printf("----------------SUCCESS!---------------\n");
 exit(0);
}
```

Sample outputs (from terminal)

Correct :

```
CD_A1@CL3-25:~/Desktop/College/Sem6/CD/Lab/Lab6$ ./q1
Enter  String:  a$
----------------SUCCESS!---------------
```

Incorrect :

```
CD_A1@CL3-25:~/Desktop/College/Sem6/CD/Lab/Lab6$ ./q1
Enter  String:  ab$
b
-----------------ERROR!----------------
```

2. Grammar (Q2)

- Original: S → U V W; U → (S) | a S b | d; V → a V | ε; W → c W | ε (already LL(1))
- LL(1) elimination: not needed

Code (q2.c)

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

void S();
void U();
void V();
void W();

void invalid();
void valid();


int curr = 0;
char str[100];

void main() {
    printf("Enter  String:  ");
    scanf("%s", str);
    S();
    if(str[curr] == '$') valid();
    else {
        printf("%c\n", str[curr]);
        invalid();
    }
}

void S(){
    if(str[curr] == 'a' || str[curr] == 'd') {
        U();
        V();
        W();
        return;
    }
    invalid();
}

void U(){
    if(str[curr] == 'a') {
        curr++;
        S();
        if(str[curr] == 'b') curr++;
        return;
    }
```

```
    if(str[curr] == 'd') {
        curr++;
        return;
    }
    if(str[curr] == '(') {
        curr++;
        S();
        if(str[curr] == ')') curr++;
        return;
    }
    invalid();
}
void V(){
    if(str[curr] == 'a') {
        curr++;
        V();
        return;
    }
    else if(str[curr] == 'b' || str[curr] == 'c' || str[curr] == ')' ||
str[curr] == '$')
        return;
    invalid();
}
void W(){
    if(str[curr] == 'c') {
        curr++;
        W();
        return;
    }
    else if(str[curr] == 'b' || str[curr] == ')' || str[curr] == '$')
        return;
    invalid();
}

void invalid() {
   printf("-----------------ERROR!----------------\n");
   exit(0);
}

void valid() {
  printf("----------------SUCCESS!---------------\n");
 exit(0);
}
```

Sample outputs (from terminal)

Correct :

```
CD_A1@CL3-25:~/Desktop/College/Sem6/CD/Lab/Lab6$ ./q2
Enter  String:  daacc$
----------------SUCCESS!---------------
```

Incorrect :

```
CD_A1@CL3-25:~/Desktop/College/Sem6/CD/Lab/Lab6$ ./q2
Enter  String:  ab$
-----------------ERROR!----------------
```

3. Grammar (Q3)

- Original: S → a A c B e; A → A b | b (left recursion); B → d
- LL(1) elimination applied on A:
  - A → b A'
  - A' → b A' | ε

Code (q3.c)

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

void S();
void A();
void Ap();
void B();

void invalid();
void valid();


int curr = 0;
char str[100];

void main() {
    printf("Enter  String:  ");
    scanf("%s", str);
    S();
    if(str[curr] == '$') valid();
    else {
        printf("%c\n", str[curr]);
        invalid();
    }
}

void S() {
    if(str[curr] == 'a') {
        curr++;
        A();
        if(str[curr] == 'c') curr++;
        B();
```

```
        if(str[curr] == 'e') curr++;
        return;
    }
    invalid();
}

void A() {
    if(str[curr] == 'b') {
        curr++;
        Ap();
        return;
    }
    invalid();
}

void Ap() {
    if(str[curr] == 'b') {
        curr++;
        Ap();
        return;
    }
    else if(str[curr] == 'c')
        return;
    invalid();
}

void B() {
    if(str[curr] == 'd') {
        curr++;
        return;
    }
    invalid();
}

void invalid() {
   printf("-----------------ERROR!----------------\n");
   exit(0);
}

void valid() {
  printf("---------------SUCCESS!---------------\n");
 exit(0);
}
```

Sample outputs (from terminal)

Correct :

```
CD_A1@CL3-25:~/Desktop/College/Sem6/CD/Lab/Lab6$ ./q3
Enter  String:  abcde$
---------------SUCCESS!---------------
```

Incorrect :

```
CD_A1@CL3-25:~/Desktop/College/Sem6/CD/Lab/Lab6$ ./q3
Enter  String:  acde$
-----------------ERROR!----------------
```

4. Grammar (Q4)

- Original: S → (L) | a; L → L , S | S (left recursion)
- LL(1) elimination applied on L:
  - L → S L'
  - L' → , S L' | ε

Code (q4.c)

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

void S(void);
void L(void);
void Lp(void);
void invalid(void);
void valid(void);

int curr = 0;
char str[100];

int main(void) {
    printf("Enter  String:  ");
    if (scanf("%99s", str) != 1) {
        invalid();
    }

    S();

    if (str[curr] == '$') {
        valid();
    } else {
        invalid();
    }

    return 0;
}

void S(void) {
    if (str[curr] == '(') {
        curr++;
```

```
        L();
        if (str[curr] == ')') {
            curr++;
            return;
        }
        invalid();
    }

    if (str[curr] == 'a') {
        curr++;
        return;
    }

    invalid();
}

void L(void) {
    if (str[curr] == '(' || str[curr] == 'a') {
        S();
        Lp();
        return;
    }

    invalid();
}

void Lp(void) {
    if (str[curr] == ',') {
        curr++;
        S();
        Lp();
        return;
    }

    if (str[curr] == ')') {
        return;
    }

    invalid();
}

void invalid(void) {
    printf("-----------------ERROR!----------------\n");
    exit(0);
}

void valid(void) {
    printf("----------------SUCCESS!---------------\n");
    exit(0);
}
```

Sample outputs (from terminal)

Correct :

```
CD_A1@CL3-25:~/Desktop/College/Sem6/CD/Lab/Lab6$ ./q4
Enter  String:  (a,a)$
----------------SUCCESS!---------------
```

Incorrect :

```
CD_A1@CL3-25:~/Desktop/College/Sem6/CD/Lab/Lab6$ ./q4
Enter  String:  (a,$
-----------------ERROR!----------------
```