**6. Implement a C program to eliminate left recursion.**

**Code:**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAX\_RULES 10

#define MAX\_LEN 100

void removeLeftRecursion(char nt, char prods[][MAX\_LEN], int count) {

char left[MAX\_RULES][MAX\_LEN];

char right[MAX\_RULES][MAX\_LEN];

int leftCount = 0, rightCount = 0;

for (int i = 0; i < count; i++) {

if (prods[i][0] == nt) {

strcpy(left[leftCount], prods[i] + 1);

leftCount++;

} else {

strcpy(right[rightCount], prods[i]);

rightCount++;

}

}

if (leftCount == 0) {

printf("%c ->", nt);

for (int i = 0; i < count; i++) {

printf(" %s", prods[i]);

if (i < count - 1) printf(" | ");

}

printf("\n");

return;

}

char newNt = nt + 1;

printf("%c ->", nt);

for (int i = 0; i < rightCount; i++) {

printf(" %s%c'", right[i], newNt);

if (i < rightCount - 1) printf(" | ");

}

printf("\n");

printf("%c' ->", newNt);

for (int i = 0; i < leftCount; i++) {

printf(" %s%c'", left[i], newNt);

printf(" | ");

}

printf(" ε\n");

}

int main() {

int num;

char nt;

char prods[MAX\_RULES][MAX\_LEN];

printf("Enter number of productions: ");

scanf("%d", &num);

for (int i = 0; i < num; i++) {

printf("Enter production %d (e.g., A->Ab|c): ", i + 1);

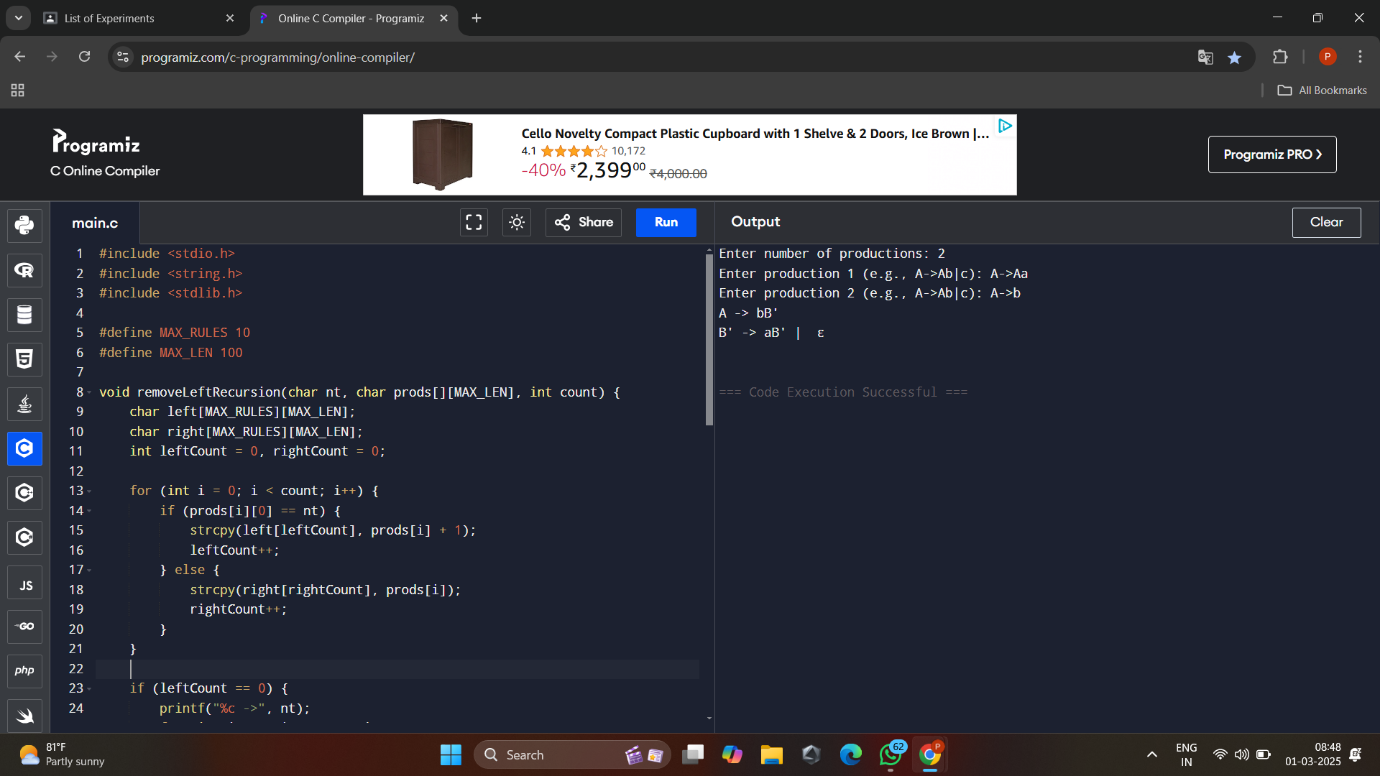
scanf(" %c->%s", &nt, prods[i]);

}

removeLeftRecursion(nt, prods, num);

return 0;

}

****