**TOC- C PROGRAM**

**1.Write a C program to simulate a Deterministic Finite Automata (DFA) for the given language representing strings that start with a and end with a**

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

scanf("%s", str);

int len = strlen(str);

printf((str[0] == 'a' && str[len-1] == 'a') ? "Accepted\n" : "Rejected\n");

return 0;

}

**Input:**

aba

**Output:**

Accepted

**2.Write a C program to simulate a Deterministic Finite Automata (DFA) for the given language representing strings that start with 0 and end with 1**

#include <stdio.h>

#include <string.h>

int main() {

char s[100]; scanf("%s", s);

int l = strlen(s);

printf((s[0] == '0' && s[l-1] == '1') ? "Accepted\n" : "Rejected\n");

}

**Input:**

01101

**Output:**

Accepted

**3. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)**

**S0A1 A 0A | 1A | ε**

#include <stdio.h>

#include <string.h>

int check(char \*s) {

int l = strlen(s);

return l >= 3 && s[0] == '0' && s[l-1] == '1';

}

int main() {

char s[100]; scanf("%s", s);

printf(check(s) ? "Accepted\n" : "Rejected\n");

}

**Input:**

0101

**Output:**

Accepted

**4. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)**

**SOSO | 1S1|0|1|ε**

#include <stdio.h>

#include <string.h>

int isPal(char \*s) {

int i = 0, j = strlen(s) - 1;

while (i < j) if (s[i++] != s[j--]) return 0;

return 1;

}

int main() {

char s[100]; scanf("%s", s);

printf(isPal(s) ? "Accepted\n" : "Rejected\n");

}

**Input:**

0110

**Output:**

Accepted

**5. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)**

**SOSOA Α1Αε**

#include <stdio.h>

#include <string.h>

int check(char \*s) {

for (int i = 0; s[i]; i++) if (s[i] != '0' && s[i] != '1') return 0;

return 1;

}

int main() {

char s[100]; scanf("%s", s);

printf(check(s) ? "Accepted\n" : "Rejected\n");

}

**Input:**

0011

**Output:**

Accepted

**6. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)**

**SOS1 | ε**

#include <stdio.h>

#include <string.h>

int main() {

char s[100]; int c0 = 0, c1 = 0;

scanf("%s", s);

for (int i = 0; s[i]; i++)

if (s[i] == '0') c0++;

else if (s[i] == '1') c1++;

printf((c0 == c1) ? "Accepted\n" : "Rejected\n");

}

**input:**

0011

**Output:**

Accepted

**7. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)**

**SA101A, ΑΠ0Α | 1A | ε**

#include <stdio.h>

#include <string.h>

int main() {

char s[100]; scanf("%s", s);

printf(strstr(s, "101") ? "Accepted\n" : "Rejected\n");

}

**Input:**

0110100

**Output:**

Accepted

**8. Write a C program to simulate a Non-Deterministic Finite Automata (NFA) for the given languagerepresenting strings that start with b and end with a**

#include <stdio.h>

#include <string.h>

int main() {

char s[100]; scanf("%s", s);

int l = strlen(s);

printf((s[0] == 'b' && s[l-1] == 'a') ? "Accepted\n" : "Rejected\n");

}

**Input:**

bba

**Output:**

Accepted

**9. Write a C program to simulate a Non-Deterministic Finite Automata (NFA) for the given languagerepresenting strings that start with o and end with 1**

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

printf("Enter a binary string: ");

scanf("%s", str);

int len = strlen(str);

if (len >= 2 && str[0] == '0' && str[len - 1] == '1')

printf("Accepted\n");

else

printf("Rejected\n");

return 0;

}

**Input:**

Enter a binary string: 011

**Output:**

Accepted

**10. Write a C program to find ɛ -closure for all the states in a Non-Deterministic Finite Automata (NFA) with ɛ -moves.**

#include <stdio.h>

int main() {

int n, e[10][10] = {0};

printf("States: "); scanf("%d", &n);

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

int x, k; printf("?-transitions from %d: ", i);

scanf("%d", &k);

while (k--) { scanf("%d", &x); e[i][x] = 1; }

}

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

printf("?-closure(%d): %d", i, i);

for (int j = 0; j < n; j++) if (e[i][j]) printf(", %d", j);

printf("\n");

}

}

**Input:**

Enter number of states: 3

Enter number of ɛ-transitions from state 0: 2

Enter ɛ-transition states: 1 2

Enter number of ɛ-transitions from state 1: 0

Enter number of ɛ-transitions from state 2: 1

Enter ɛ-transition states: 1

**Output:**

ɛ-closure(0): 0, 1, 2

ɛ-closure(1): 1

ɛ-closure(2): 2, 1

**11. Write a C program to find e -closure for all the states in a Non-Deterministic Finite Automata (NFA) with ɛ -moves.**

#include <stdio.h>

int main() {

int n, e[10][10] = {0};

printf("States: "); scanf("%d", &n);

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

int x, k; printf("?-transitions from %d: ", i);

scanf("%d", &k);

while (k--) { scanf("%d", &x); e[i][x] = 1; }

}

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

printf("?-closure(%d): %d", i, i);

for (int j = 0; j < n; j++) if (e[i][j]) printf(", %d", j);

printf("\n");

}

}

**Input:**

Enter number of states: 3

Enter number of ɛ-transitions from state 0: 2

Enter ɛ-transition states: 1 2

Enter number of ɛ-transitions from state 1: 0

Enter number of ɛ-transitions from state 2: 1

Enter ɛ-transition states: 1

**Output:**

ɛ-closure(0): 0, 1, 2

ɛ-closure(1): 1

ɛ-closure(2): 2, 1