**main.c** `[ ]` `☼` `⤴ Share` `Run`

**Output**

```c
1  #include <stdio.h>
2  #include <string.h>
3  int main() {
4      char str[100];
5      printf("Enter string: ");
6      scanf("%s", str);
7
8      int len = strlen(str);
9      if (len >= 2 && str[0] == 'a' && str[len - 1] == 'a')
10         printf("Accepted\n");
11     else
12         printf("Rejected\n");
13
14     return 0;
15 }
```

```
Enter string: aabb
Rejected


=== Code Execution Successful ===
```

```c
1   #include <stdio.h>
2   #include <string.h>
3 ▾ int main() {
4       char str[100];
5       printf("Enter string: ");
6       scanf("%s", str);
7
8       int len = strlen(str);
9       if (len >= 2 && str[0] == 'o' && str[len - 1] == '1')
10          printf("Accepted (NFA Simulation)\n");
11      else
12          printf("Rejected\n");
13
14      return 0;
15  }
16
```

**Output**

```
Enter string: o0111
Accepted (NFA Simulation)


=== Code Execution Successful ===
```

```c
#include <stdio.h>
#define STATES 3
int eClosure[STATES][STATES] = {
    {1, 1, 0},
    {0, 1, 1},
    {0, 0, 1}
};
void findEClosure(int state) {
    printf("ε-closure(%d): {", state);
    for (int i = 0; i < STATES; i++) {
        if (eClosure[state][i])
            printf(" %d", i);
    }
    printf(" }\n");
}

int main() {
    for (int i = 0; i < STATES; i++) {
        findEClosure(i);
    }
    return 0;
}
```

Output

```
ε-closure(0): { 0 1 }
ε-closure(1): { 1 2 }
ε-closure(2): { 2 }


=== Code Execution Successful ===
```
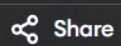
```c
#include <stdio.h>
#include <string.h>
int checkA(const char *str, int i, int len) {
    while (i < len - 1) {
        if (str[i] != '0' && str[i] != '1') return 0;
        i++;
    }
    return 1;
}
int main() {
    char str[100];
    printf("Enter string: ");
    scanf("%s", str);
    int len = strlen(str);

    if (len >= 2 && str[0] == '0' && str[len - 1] == '1' && checkA(str
        , 1, len))
        printf("Accepted\n");
    else
        printf("Rejected\n");

    return 0;
}
```

Output:

Enter string: 0111
Accepted

=== Code Execution Successful ===

```c
#include <stdio.h>
#include <string.h>
int check(const char *str, int start, int end) {
    if (start > end) return 1;
    if (start == end) return str[start] == '0' || str[start] == '1';
    if (str[start] == str[end]) return check(str, start + 1, end - 1);
    return 0;
}

int main() {
    char str[100];
    printf("Enter string: ");
    scanf("%s", str);
    int len = strlen(str);
    if (check(str, 0, len - 1))
        printf("Accepted\n");
    else
        printf("Rejected\n");
    return 0;
}
```

Output

```
Enter string: 010
Accepted

=== Code Execution Successful ===
```

**main.c**     [ ]   ☀   ⌔ Share   **Run**

```c
#include <stdio.h>
#include <string.h>
int checkA(const char *str, int i, int len) {
    for (; i < len; i++)
        if (str[i] != '1') return 0;
    return 1;
}
int checkS(const char *str, int start, int end) {
    if (start > end) return 0;
    if (str[start] == '0' && str[end] == '0') return checkS(str, start
        + 1, end - 1);
    return checkA(str, start, end + 1);
}
int main() {
    char str[100];
    printf("Enter string: ");
    scanf("%s", str);
    int len = strlen(str);
    if (checkS(str, 0, len - 1))
        printf("Accepted\n");
    else
        printf("Rejected\n");
    return 0;
}
```

**Output**

Enter string: 010
Accepted

=== Code Execution Successful ===

```c
main.c

1   #include <stdio.h>
2   #include <string.h>
3
4   int isValid(const char *str, int start, int end) {
5       if (start > end) return 1;
6       if (str[start] == '0' && str[end] == '1') return isValid(str,
            start + 1, end - 1);
7       return 0;
8   }
9
10  int main() {
11      char str[100];
12      printf("Enter string: ");
13      scanf("%s", str);
14      int len = strlen(str);
15      if (isValid(str, 0, len - 1))
16          printf("Accepted\n");
17      else
18          printf("Rejected\n");
19      return 0;
20  }
21
```

Output

```
Enter string: 0111
Rejected


=== Code Execution Successful ===
```

```c
#include <stdio.h>
#include <string.h>
int isA(const char *str, int start, int end) {
    for (int i = start; i <= end; i++) {
        if (str[i] != '0' && str[i] != '1') return 0;
    }
    return 1;
}
int main() {
    char str[100];
    printf("Enter string: ");
    scanf("%s", str);
    int len = strlen(str);

    for (int i = 0; i <= len - 5; i++) {
        if (str[i + 1] == '1' && str[i + 2] == '0' && str[i + 3] ==
            '1') {
            if (isA(str, 0, i) && isA(str, i + 4, len - 1)) {
                printf("Accepted\n");
                return 0;
            }
        }
    }
    printf("Rejected\n");
```

**Output**

```
Enter string: 1111
Rejected


=== Code Execution Successful ===
```