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

S → 0S0 | A A → 1A | ε

Aim: To write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

S → 0S0 | A A → 1A | ε

Algorithm:

1. Get the input string from the user.

2. Find the length of the string.

3. Check whether all the symbols in the input are either 0 or 1. If so, print “String is valid” and go to step 4. Otherwise print “String not valid” and quit the program.

4. Read the input string character by character

5. Count the number of 0’s in the front and store it in the variable count1

6. Skip all 1’s

7. Count the number of 0’s in the end and store it in the variable count2

8. If count1==count2, print “String Accepted”. Otherwise print “String Not Accepted”

Program:

#include<stdio.h>

#include<string.h>

void main()

{

char s[100];

int i,flag,flag1,a,b;

int l,count1,count2;

printf("enter a string to check:");

scanf("%s",s);

l=strlen(s);

flag=1;

for(i=0;i<l;i++)

{

if(s[i]!='0' && s[i]!='1')

{

flag=0;

}

}

if(flag!=1)

printf("string is Not Valid\n");

if(flag==1)

{

i=0;count1=0;

while(s[i]=='0') // Count the no of 0s in the front

{

count1++;

i++;

}

while(s[i]=='1')

{

i++; // Skip all 1s

}

flag1=1;

count2=0;

while(i<l)

{

if(s[i]=='0')// Count the no of 0s at the end

{

count2++;

}

else

{

flag1=0;

}

i++;

}

if(flag1==1)

{

if(count1==count2)

{

printf("The string satisfies the condition 0n1m0n\n");

printf("String Accepted\n");

}

else

{

printf("The string does not satisfy the condition 0n1m0n\n");

printf("String Not Accepted\n");

}

}

else

{

printf("The string does not satisfy the condition 0n1m0n\n");

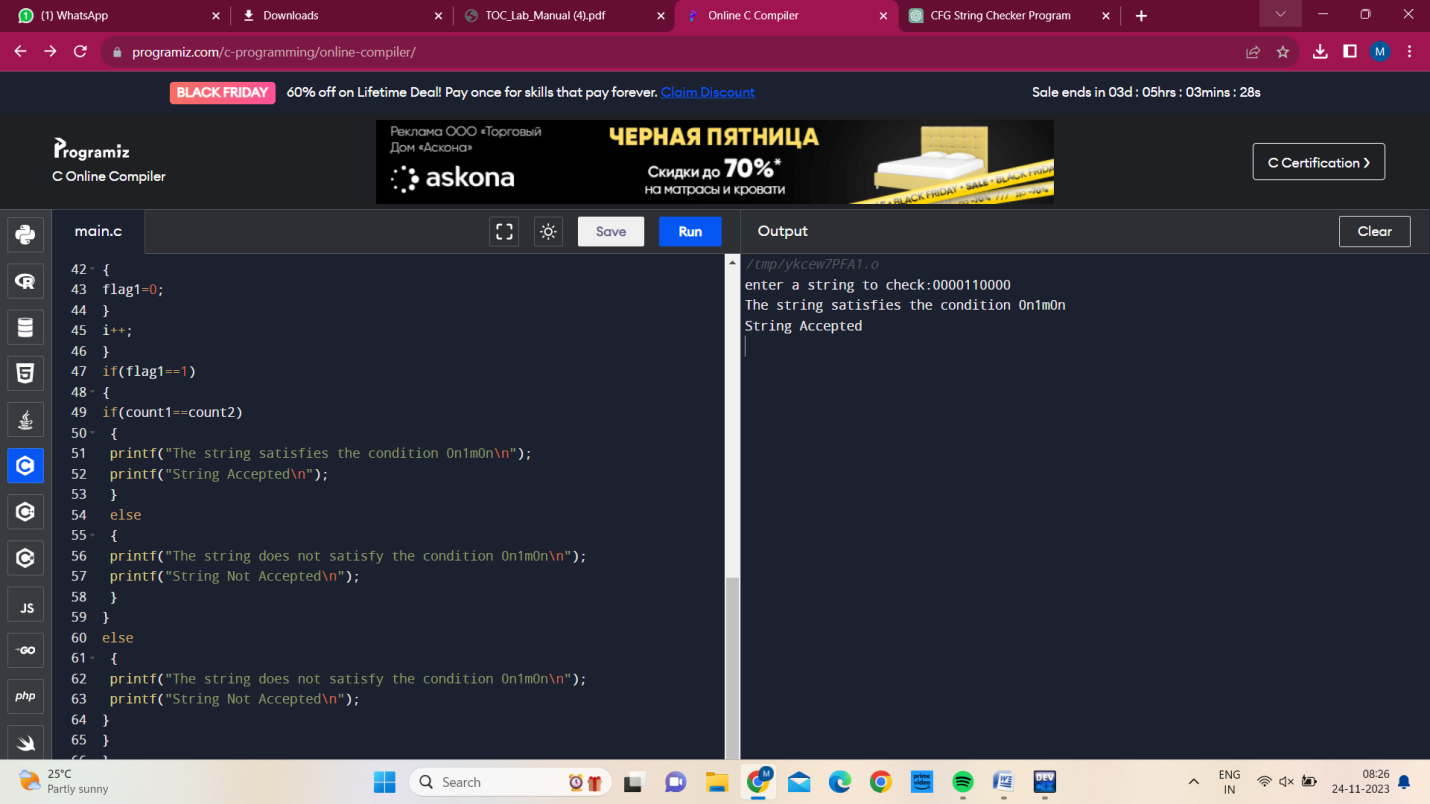
printf("String Not Accepted\n");

}

}

}

Output:



Result:

C program is executed successfully.