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

S → 0S1 | ε

AIM : To write a C program to check whether a string belongs to the grammar S -> 0 S 1 | ε Language defined by the Grammar Set of all strings over 𝚺={0,1} satisfying 0n1n

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. Find the length of the string. If the length is odd, then print “String not accepted” and quit the program. If the length is even, then go to step 5.

5. Divide the string into two halves.

6. If the first half contains only 0s and the second half contains only 1s then print “String Accepted”. Otherwise print “String Not Accepted

PROGRAM:

#include<stdio.h>

#include<string.h>

void main()

{

char s[100];

int i,flag,flag1,flag2;

int l;

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)

{

if(l%2!=0) // If string length is odd

{

printf("The string does not satisfy the condition 0n1n\n");

printf("String Not Accepted\n");

}

else

{

// To check first half contains 0s

flag1=1;

for(i=0;i<(l/2);i++)

{

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

{

flag1=0;

}

}

// To check second half contains 1s

flag2=1;

for(i=l/2;i<l;i++)

{

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

{

flag2=0;

}

}

if(flag1==1 && flag2==1)

{

printf("The string satisfies the condition 0n1n\n");

printf("String Accepted\n");

}

else

{

printf("The string does not satisfy the condition 0n1n\n");

printf("String Not Accepted\n");

}

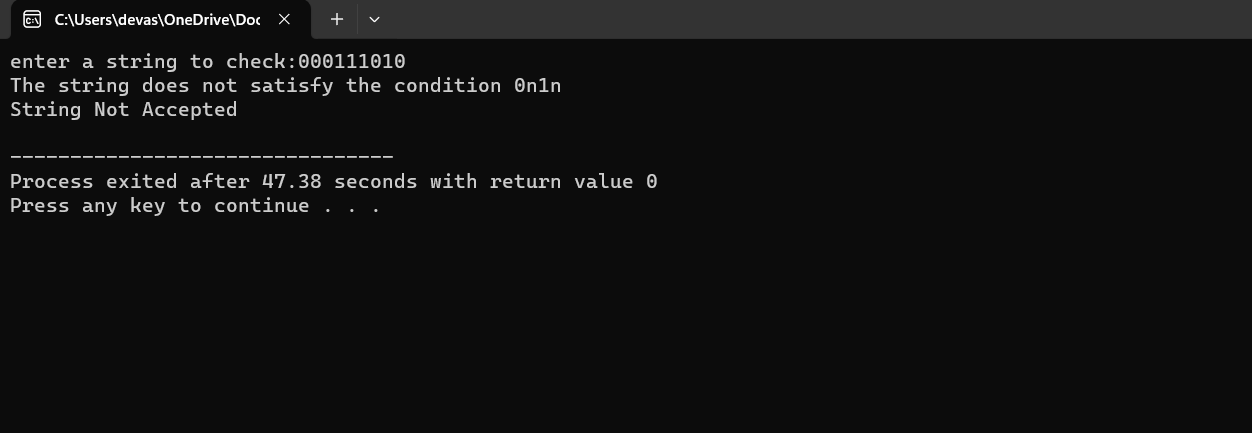
}

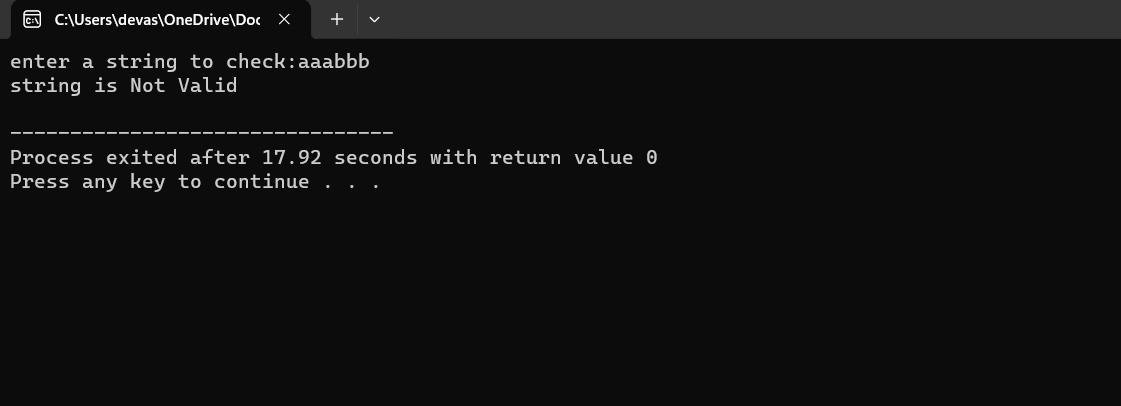
}

}

OUTPUT:







RESULT:

Thus we have successfully executed the language defined by CFG using DevC++.