

To write a C program to simulate a Turing Machine for the language $L = \{ 0^n 1^{2n} \mid n \geq 1 \}$

$1^{2n} \mid n \geq 1 \}$ in which n number of 0's are followed by 2n number of 1's

AIM : To write a C program to simulate a Turing Machine for the language $L = \{ 0^n 1^{2n} \mid n \geq 1 \}$ in which n number of 0's are followed by 2n number of 1's

Program:

```
#include<stdio.h>

#include<string.h>

void main()

{

int i,j,le,flag,flag1,flag2;

char str[20];

printf("Program to show how a turing machine will process 0n1n2n\n");

printf("Enter a string : ");

scanf("%s",str);

le=strlen(str);

j=0;

while(1)

{

flag=0;flag1=0;flag2=0;i=0;

while(i<le)

{

if((str[i]=='0')&&(flag==0))

{

str[i] = 'A';

printf("%s\n",str);

flag=1; //To mark that a 0 is changed to A

i=i+1;

}

else if((str[i]=='0')&&(flag==1))

{

i=i+1; //Skip 0
```

```
}  
else if(str[i]=='A')  
{  
i=i+1; //Skip A  
}  
else if((str[i]=='1')&&(flag1==0))  
{  
str[i] = 'B';  
printf("%s\n",str);  
flag1=1; //To mark that a 1 is changed to B  
i=i+1;  
}  
else if((str[i]=='1')&&(flag1==1))  
{  
i=i+1; //Skip 1  
}  
else if(str[i]=='B')  
{  
i=i+1; //Skip B  
}  
else if((str[i]=='2')&&(flag2==0))  
{  
str[i] = 'C';  
printf("%s\n",str);  
flag2=1; //To mark that a 2 is changed to C  
i=i+1;  
}  
else if((str[i]=='2')&&(flag2==1))  
{  
i=i+1; //Skip 2  
}
```

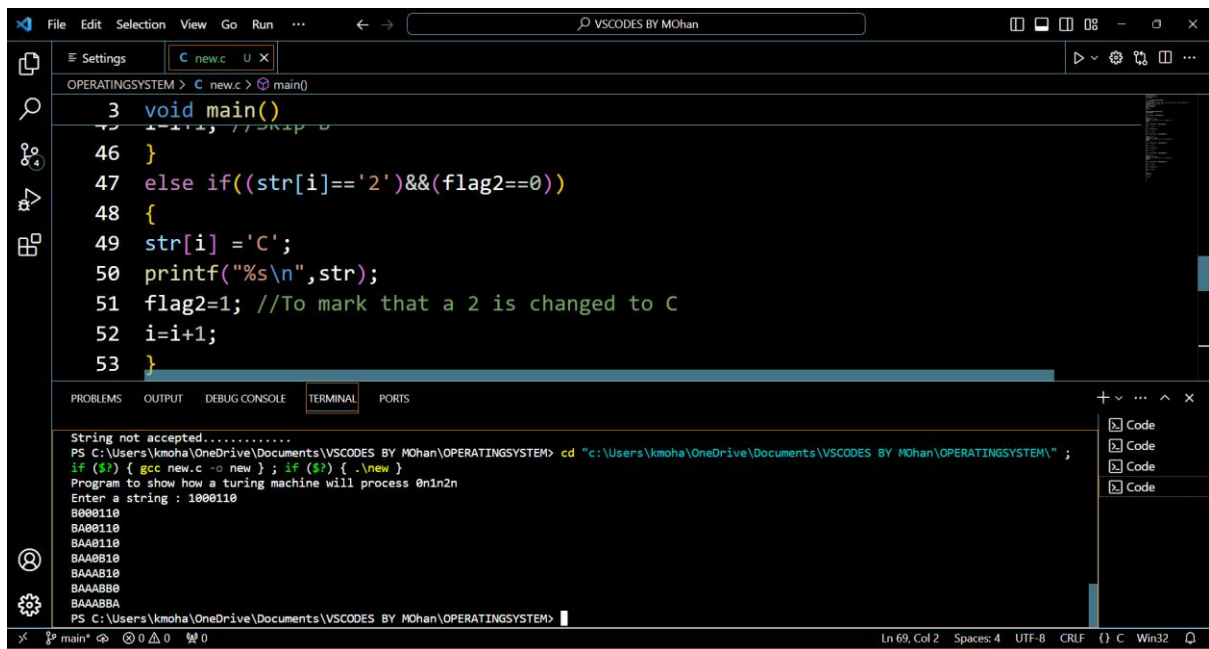
```

else if(str[i]=='C')
{
i=i+1; //Skip C
}
}

j=j+1;
if(j==le)
{
break;
}
}
}}

```

OUTPUT:



The screenshot shows the VS Code interface with a C program in the editor and its execution output in the terminal. The program is a Turing machine simulation that processes a binary string. The terminal output shows the program's execution, including the input string '1000110' and the resulting output 'BAA0110'.

```

3 void main()
4 {
5     char str[100];
6     int i=0, j=0, le=0, flag2=0;
7     printf("Enter a string : ");
8     while(str[i]!='\n')
9     {
10         str[i] = getche();
11         i++;
12     }
13     str[i] = '\n';
14     le = i;
15     while(j<le)
16     {
17         if(str[j]=='0')
18         {
19             if(flag2==0)
20             {
21                 str[j] = 'B';
22                 printf("%s\n", str);
23                 flag2=1; //To mark that a 0 is changed to B
24                 i=i+1;
25             }
26             else if(str[i]=='C')
27             {
28                 str[i] = 'A';
29                 printf("%s\n", str);
30                 flag2=1; //To mark that a C is changed to A
31                 i=i+1;
32             }
33             else if(str[i]=='B')
34             {
35                 str[i] = 'A';
36                 printf("%s\n", str);
37                 flag2=1; //To mark that a B is changed to A
38                 i=i+1;
39             }
40             else if(str[i]=='A')
41             {
42                 str[i] = 'B';
43                 printf("%s\n", str);
44                 flag2=1; //To mark that a A is changed to B
45                 i=i+1;
46             }
47         }
48         j++;
49     }
50     printf("String not accepted.....\n");
51 }

```

```

PS C:\Users\kmoha\OneDrive\Documents\VSCODES BY MOhan\OPERATINGSYSTEM> cd "c:\Users\kmoha\OneDrive\Documents\VSCODES BY MOhan\OPERATINGSYSTEM\" ;
if ($?) { gcc new.c -o new } ; if ($?) { .\new }
Program to show how a turing machine will process 0n1n2n
Enter a string : 1000110
B000110
BA00110
BAA0110
BAA0B10
BAAAB10
BAAABBB0
BAAABBA
PS C:\Users\kmoha\OneDrive\Documents\VSCODES BY MOhan\OPERATINGSYSTEM>

```

RESULT: Output is successfully obtained.