

To write a C program to simulate a PDA 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

AIM : To write a C program to simulate a PDA 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>

char stack[20];

int top,count=0;

void push()
{
    top=top+1;
    stack[top]='0';
    stack[top+1]='\0';
}

int pop()
{
    if(top<1)
        return(0);
    else
    {
        stack[top]='\0';
        top=top-1;
        return(1);
    }
}

void main()
{
    int m,i,j,k,l,a,len;
    char input[20],rem_input[20];
    printf("Simulation of PDA for n 0's followed by 2n 1's\n");
```

```

printf("Enter a string : ");
scanf("%s",input);
l=strlen(input);
j=0;stack[0]='Z';top=0;
printf("Stack\tInput\n");
printf("%s\t%s\n",stack,input);
while(1)
{
len=strlen(input);
while(len>0)
{
if(input[0]=='0')
{
push();
m=0;
for(k=1;k<len;k++)
{
rem_input[m]=input[k];
m=m+1;
}
rem_input[m]='\0';
strcpy(input,rem_input);
printf("%s\t%s\n",stack,input);
}
if(input[0]=='1')
{
count++;
if(count%2==0)
{
a=pop();
if(a==0)

```

```
{  
printf("String not accepted");  
goto b;  
}  
  
else  
  
{  
m=0;  
for(k=1;k<len;k++)  
{  
rem_input[m]=input[k];  
m=m+1;  
}  
}  
rem_input[m]='\0';  
strcpy(input,rem_input);  
printf("%s\t%s\n",stack,input);  
}  
  
else  
  
{  
m=0;  
for(k=1;k<len;k++)  
{  
rem_input[m]=input[k];  
m=m+1;  
}  
rem_input[m]='\0';  
strcpy(input,rem_input);  
printf("%s\t%s\n",stack,input);  
}  
}  
  
break;
```

```

}

j=j+1;

//printf("j = %d\t l = %d\n",j,l);

if(j==l)

{

break;

}

}

if(top>=1)

{

printf("String not accepted");

}

else

{

printf("String accepted");

}

b:

printf(".....");

```

OUTPUT:

The screenshot shows a Visual Studio Code editor window with a C program in a file named 'new.c'. The program is a stack-based string reversal algorithm. The code is as follows:

```

22 void main()
36 while(len>0)
37 {
38     if(input[0]!='\0')
39     {
40         push();
41         m=0;
42         for(k=1;k<len;k++)
43         {
44             rem input[m]=input[k];

```

The terminal output shows the program's execution. It prompts the user to enter a string, and the input '010011001' is shown. The stack is then displayed, showing the characters of the string being pushed onto it. The output 'String not accepted.....' is shown at the bottom of the terminal.

Stack Input

Address	Value
Z	010011001
Z0	10011001
Z0	0011001
Z00	011001
Z000	11001
Z00	1001
Z00	001
Z000	01
Z0000	1

String not accepted.....

}RESULT: Output is successfully obtained.