**AIM :**

To write a C program to simulate a PDA for the language L={ 0n 12n | n>=1 }

in which *n* number of 0’s are followed by *2n* number of 1’s

**ALGORITHM :**

1. Get the input string from the user.

2. Define a stack and push the symbol ‘Z’ onto the stack. The symbol ‘Z’

acts as the bottom marker of the stack.

3. Initialize a variable *count=0*

4. Find the length of the string.

5. Read the input string character by character.

6. Read the current input symbol do steps 7 and 8. If the end of the

input is reached, go to step 9

7. If the input symbol is 0, push it onto the stack. Print the content of

the stack and the remaining input and go to step 6

8. If the input symbol is 1

a. Increment *count*.

b. If *count* is odd, go to step 6 to read the next input symbol

c. If *count* is even, check whether there is a 0 at the top of the

stack. If so, pop it from the stack. Print the content of the stack

and the remaining input and go to step 6. If not, print “String

not accepted” and quit the program

9. If the stack is empty having only the bottom marker, print “String

Accepted”. Otherwise print “String not accepted”.

**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");

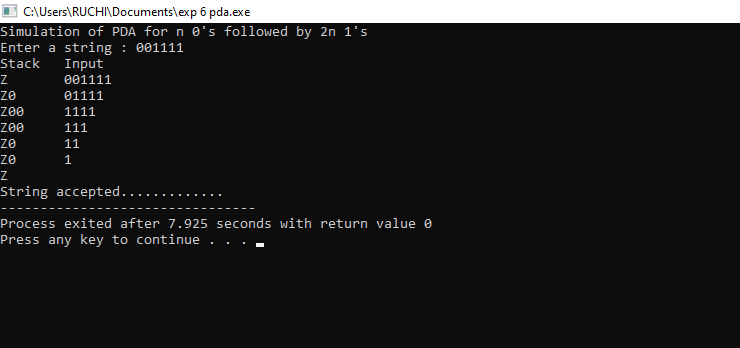
}

b:

printf(".............");

}

**OUTPUT :**



RESULT:

Thus executed successfully