

```
#include <iostream>
#include <fstream>
using namespace std;
```

```
int ESPR_size = 0;
```

```
ifstream finp ("parentin.txt");
```

```
struct Nodo {
    char ESPR;
    Nodo* pSucc;
};
Nodo* pTesta = NULL;
Nodo* pNew = NULL;
Nodo* pTemp;
```

```
int Push (char let){
    pNew = new Nodo;
    pNew -> ESPR = let;
    pNew -> pSucc = pTesta;
    pTesta = pNew;
    ESPR_size++;
    cout << ESPR_size << "\tINS " << let << '\t' << pTesta << endl;
}
```

```
int Pop (char& LetTesta){
    if ( pTesta == NULL ) {
        cout<<"\n\n errore: Pila vuota";
        return -1;    }
    LetTesta = pTesta -> ESPR;
    pTemp = pTesta -> pSucc;
    cout<< ESPR_size <<"\tCAN"<< LetTesta <<'\t'<< pTesta << endl;
    delete pTesta;
    pTesta = pTemp;
    ESPR_size--;
    return 0;
}
```

```
int main(){
    char LET, letPila;
    int N, esito=0;
    finp >> N;
    for (int i=0; i<N && esito==0; i++){
        finp >> LET;
        if (LET == '{' || LET == '['
            || LET == '(' || LET == '<')
            Push(LET);
        else
        {   esito = Pop (letPila);
            if (esito == 0)
                if (LET == '}' && letPila == '{'
                    || LET == ']' && letPila == '['
                    || LET == ')' && letPila == '('
                    || LET == '>' && letPila == '<')
                    esito = 0;
                else
                    esito = -1;    }
        }
    if ( esito < 0)
        cout << "\n\n sequenza errata";
    else
        if ( pTesta != NULL ) {
            cout << "\n\n sequenza errata, Pila piena, n.elem= ";
            cout << ESPR_size << "\n\npTesta: \t"<< pTesta <<endl;
        }
        else
            cout << "\n\n sequenza corretta";
    finp.close();
}
```

parentin.txt :

4
{[]

```
>>
>>parentesi_puntatori.exe
1      INS (    0x7b9ee8
2      INS {    0x7b9f18
3      INS [    0x7b9f68
3      CAN [    0x7b9f68
```

sequenza errata, Pila piena, n.elem= 2

pTesta: 0x7b9f18

>>

parentin.txt :

4
)[](

```
>>
>>parentesi_puntatori.exe
```

errore: Pila vuota

sequenza errata
>>

parentin.txt :

26

{[((<{[((<{[((<())>)]}>)]}>)]}

>>

>>parentesi_puntatori.exe

1	INS	{	0xc39fe8
2	INS	[0xc39f28
3	INS	(0xc39ee8
4	INS	<	0xc39ef8
5	INS	{	0xc39fa8
6	INS	[0xc3a008
7	INS	(0xc39f08
8	INS	<	0xc39fb8
9	INS	{	0xc39fd8
10	INS	[0xc39f18
11	INS	(0xc39f38
12	INS	<	0xc39f58
13	INS	(0xc39ff8
13	CAN	(0xc39ff8
12	CAN	<	0xc39f58
11	CAN	(0xc39f38
10	CAN	[0xc39f18
9	CAN	{	0xc39fd8
8	CAN	<	0xc39fb8
7	CAN	(0xc39f08
6	CAN	[0xc3a008
5	CAN	{	0xc39fa8
4	CAN	<	0xc39ef8
3	CAN	(0xc39ee8
2	CAN	[0xc39f28
1	CAN	{	0xc39fe8

sequenza corretta

>>

parentin.txt :

10
(){}()[]}

```
>>
>>parentesi_puntatori.exe
1      INS (    0x7da038
1      CAN (    0x7da038
1      INS {    0x7da078
2      INS (    0x7d9ec8
2      CAN (    0x7d9ec8
2      INS [    0x7d9fd8
3      INS (    0x7d9f28
3      CAN (    0x7d9f28
2      CAN [    0x7d9fd8
1      CAN {    0x7da078
```

sequenza corretta
>>

parentin.txt :

4
([])

```
>>
>>parentesi_puntatori.exe
1      INS (    0xc09f88
2      INS [    0xc0a008
2      CAN [    0xc0a008
```

sequenza errata
>>

parentin.txt :

12
([{()()}](){

```
>>
>>parentesi_puntatori.exe
1      INS  (    0xc19ec8
2      INS  [    0xc19fb8
3      INS  {    0xc19ed8
4      INS  (    0xc19fc8
4      CAN  (    0xc19fc8
4      INS  (    0xc19f18
4      CAN  (    0xc19f18
3      CAN  {    0xc19ed8
2      CAN  [    0xc19fb8
2      INS  (    0xc1a058
2      CAN  (    0xc1a058
2      INS  {    0xc19fd8
```

sequenza errata, Pila piena, n.elem= 2

pTesta: 0xc19fd8

>>