

```

1  #include <iostream>
2  #include <windows.h>
3  #include <cstdio>
4  #include "Pila.h"
5  using namespace std;
6  int main()
7  {
8  int res = -1;
9  int x = 0;
10 int longitud = 0;
11 char expresion[1024];
12 char result[1024];
13 Pila objectPila;
14 objectPila.Inicializa();
15 do
16 {
17 system("cls");
18 cout<<"Ingresa la operacion en notacion infija: ";
19 cin.ignore();
20 gets(expresion);
21 longitud = strlen(expresion);
22 for(int y = 0; y <= longitud; y++)
23 {
24 switch(expresion[y])
25 {
26 case '(':
27 objectPila.Push(expresion[y]);
28 break;
29 case ')':
30 for(int z = objectPila.getLastPos(); z != -1; z--)
31 {
32 if(objectPila.Top() == '(')
33 {
34 objectPila.Pop();
35 z = -1;
36 break;
37 }
38 else
39 {
40 res++;
41 result[res] = objectPila.Top();
42 objectPila.Pop();
43 }
44 }
45 break;
46 case '+':
47 if(objectPila.Vacia() == 1)
48 {
49 objectPila.Push(expresion[y]);
50 break;
51 }
52 if(objectPila.Top() == '(')
53 {
54 objectPila.Push(expresion[y]);
55 break;
56 }
57 for(int z = objectPila.getLastPos(); z != -1; z--)
58 {
59 if(objectPila.Top() == '(')
60 {
61 objectPila.Pop();
62 z = -1;
63 break;
64 }
65 else
66 {

```

```

67  res++;
68  result[res] = objectPila.Top();
69  objectPila.Pop();
70  }
71  }
72  case '-':
73  if(objectPila.Vacia() == 1)
74  {
75  objectPila.Push(expresion[y]);
76  break;
77  }
78  if(objectPila.Top() == '(')
79  {
80  objectPila.Push(expresion[y]);
81  break;
82  }
83  for(int z = objectPila.getLastPos(); z != -1; z--)
84  {
85  if(objectPila.Top() == '(')
86  {
87  objectPila.Pop();
88  z = -1;
89  break;
90  }
91  else
92  {
93  res++;
94  result[res] = objectPila.Top();
95  objectPila.Pop();
96  }
97  }
98  case '*':
99  if(objectPila.Vacia() == 1)
100 {
101 objectPila.Push(expresion[y]);
102 break;
103 }
104 if(objectPila.Top() == '(')
105 {
106 objectPila.Push(expresion[y]);
107 break;
108 }
109 if(objectPila.Top() == '^' or objectPila.Top() == '*' or objectPila.Top() == '/')
110 {
111 res++;
112 result[res] = objectPila.Top();
113 objectPila.Pop();
114 objectPila.Push(expresion[y]);
115 }
116 else
117 {
118 objectPila.Push(expresion[y]);
119 }
120 break;
121 case '/':
122 if(objectPila.Vacia() == 1)
123 {
124 objectPila.Push(expresion[y]);
125 break;
126 }
127 if(objectPila.Top() == '(')
128 {
129 objectPila.Push(expresion[y]);
130 break;
131 }
132 if(objectPila.Top() == '^' or objectPila.Top() == '*' or objectPila.Top() == '/')

```

```

133 {
134     res++;
135     result[res] = objectPila.Top();
136     objectPila.Pop();
137     objectPila.Push(expresion[y]);
138 }
139 else
140 {
141     objectPila.Push(expresion[y]);
142 }
143 break;
144 case '^':
145     if(objectPila.Vacia() == 1)
146     {
147         objectPila.Push(expresion[y]);
148         break;
149     }
150     if(objectPila.Top() == '(')
151     {
152         objectPila.Push(expresion[y]);
153         break;
154     }
155     if(objectPila.Top() == '^')
156     {
157         res++;
158         result[res] = objectPila.Top();
159         objectPila.Pop();
160         objectPila.Push(expresion[y]);
161     }
162     else
163     {
164         objectPila.Push(expresion[y]);
165     }
166     break;
167 default:
168     res++;
169     result[res] = expresion[y];
170     break;
171 }
172 }
173 for(int z = objectPila.getLastPos(); z != -1; z--)
174 {
175     res++;
176     result[res] = objectPila.Top();
177     objectPila.Pop();
178 }
179 cout<<"La conversion a notacion posfija es: ";
180 for(int z = 0; z <= res; z++)
181     cout<<result[z];
182 cout<<"\n";
183 system("pause");
184 res = -1;
185 }while(x == 0);
186 }

```