***Chuyển từ biểu thức trung tố sang biểu thức hậu tố***

*–       Nếu là toán hạng: cho ra output.*

*–       Nếu là dấu mở ngoặc “(“: cho vào stack*

*–       Nếu là dấu đóng ngoặc “)”: lấy các toán tử trong stack ra và cho vào output cho đến khi gặp dấu mở ngoặc “(“. (Dấu mở ngoặc cũng phải được đưa ra khỏi stack)*

*–       Nếu là toán tử:*

* *Chừng nào ở đỉnh stack là toán tử và toán tử đó có độ ưu tiên****lớn hơn hoặc bằng****toán tử hiện tại thì lấy toán tử đó ra khỏi stack và cho ra output.*
* *Đưa toán tử hiện tại vào stack*

*Sau khi duyệt hết biểu thức infix, nếu trong stack còn phần tử thì lấy các token trong đó ra và cho lần lượt vào output.*

Hãy xem ví dụ sau để hiểu rõ hơn thuật toán này.

Chúng ta sẽ chuyển biểu thức **A\*B+C\*((D-E)+F)/G** từ dạng Infix sang dạng Postfix:

|  |  |  |
| --- | --- | --- |
| **Token** | **Stack** | **Output** |
| A | {Empty} | A |
| \* | \* | A |
| B | \* | A B |
| + | + | A B \* |
| C | + | A B \* C |
| \* | + \* | A B \* C |
| ( | + \* ( | A B \* C |
| ( | + \* ( ( | A B \* C |
| D | + \* ( ( | A B \* C D |
| – | + \* ( ( – | A B \* C D |
| E | + \* ( ( – | A B \* C D E |
| ) | + \* ( | A B \* C D E – |
| + | + \* ( + | A B \* C D E – |
| F | + \* ( + | A B \* C D E – F |
| ) | + \* | A B \* C D E – F + |
| / | + / | A B \* C D E – F + \* |
| G | + / | A B \* C D E – F + \* G |
|  | {Empty} | A B \* C D E – F + \* G / + |