# Exercice sur feuille - ?valuation d'une expression postfixe-corrig?

September 18, 2020

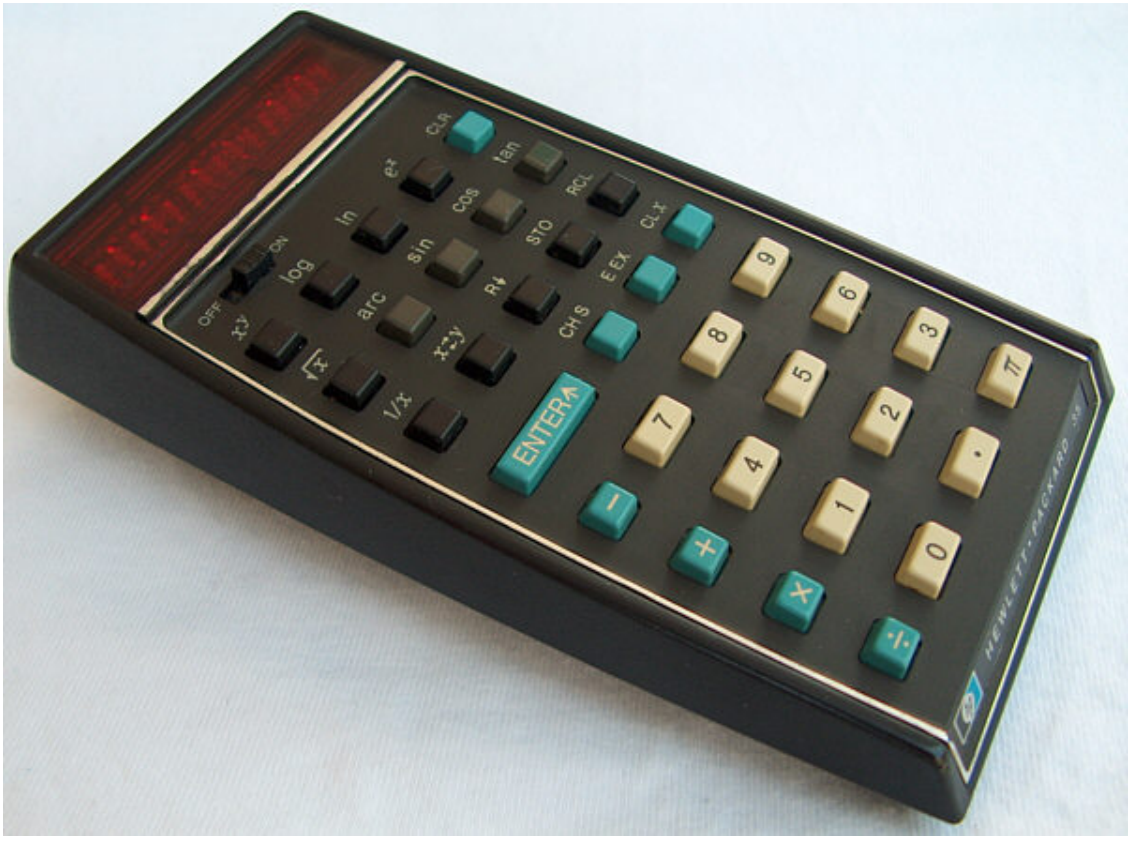
## 1 Évaluation d'une expression postfixe

Vous réaliserez l'exercice suivant d'abord sur une feuille :

- Notez vos réponses à toutes les questions sur votre feuille,
- Puis relisez-vous à l'aide des programmes déjà réalisés dans le Notebook précédent
- Quand vous êtes sûr de vous, testez vos réponses en les compilant.

[1]:
```
from IPython.display import Image
Image("img/HP_35.jpg")
```

[1]:



Cette calculatrice des années 70 fonctionne de manière particulière, elle utilise une notation appelée notation polonaise inverse pour évaluer des expressions mathématiques.

L'intérêt d'une telle notation est d'écrire une formule arithmétique sans parenthèses, on remarquera que les touches '(' et ')' sont absentes du clavier de la calculatrice.

Par exemple la formule arithmétique :

$$\frac{3+2*\sqrt{15}}{6}$$

peut s'écrire à l'aide de parenthèse comme :

$$\left(3+2*\sqrt{15}\right)/6$$

et s'écrit en notation polonaise inverse comme :

$$15, \sqrt{}, 2, *, 3, +, 6, /$$

Pour comprendre cette notation, prenons un exemple plus simple avec :

$$((3+2)*15)+6$$

Puis il faut lire l'expression dans l'ordre de ce que l'on doit faire pour réaliser le calcul : - je mets 3 (3) - j'ajoute 2 (3,2+) - je multiplie le résultat par 15 (3,2,+,15,*) - j'ajoute 6 au résultat (3,2,+,15,*,6,+)

On a donc obtenu l'expression :

$$((3+2)*15)+6 \iff 3, 2, +, 15, *, 6, +$$

En suivant le processus inverse avec

$$15, \sqrt{}, 2, *, 3, +, 6, /$$

on lit : - je prends la racine de 15 (15, $\sqrt{}$) donc $\sqrt{15}$ - je multiplie le résultat par 2 (2, *) donc $\sqrt{15}*2$ - j'ajoute 3 au résultat (3, +) donc $\sqrt{15}*2+3$ - je divise par 6 le résultat donc $\left(\sqrt{15}*2+3\right)/6$

On retrouve bien l'expression recherchée.

Une autre façon d'écrire cette expression en notation polonaise inverse est :

$$3, 2, 15, \sqrt{}, *, +, 6, /$$

cette expression est évaluée selon les étapes suivantes: - 3, 2, 15, $\sqrt{}$, , +, 6, / - 3, 2, $\sqrt{15}$, , +, 6, / - 3, $2\sqrt{15}$, +, 6, / - 3+$2\sqrt{15}$, 6, / - $\left(3+2*\sqrt{15}\right)/6$

la façon dont on lit ces différentes étapes est la suivante: - 3 est suivi d'un nombre donc je mets 3 - 2 est suivi d'un nombre donc je mets 2 - 15 est suivi d'une racine donc je calcule $\sqrt{15}$ - $\sqrt{15}$ est suivi d'une multiplication donc je le multiplie 2 par $\sqrt{15}$ donc je calcule $2*\sqrt{15}$ - $2*\sqrt{15}$ est suivi de + donc j'ajoute $2*\sqrt{15}$ à 3 donc je calcule $3+2*\sqrt{15}$ - $3+2*\sqrt{15}$ est suivi d'un nombre donc je met $3+2*\sqrt{15}$ - 6 est suivi de / donc je divise $3+2*\sqrt{15}$ par 6 donc je calcule $\left(3+2*\sqrt{15}\right)/6$

les opérations commutatives additions et multiplications permettent d'écrire de plusieurs manières différentes cette expression en notation polonaise inverse comme en notation habituelle.

### 1.0.1 Questions

1. Écrire en notation polonaise inverse les deux formes de la même expression usuelle pour les cas suivants:
   - la commutativité avec “a + b” et “b + a”
   - l’associativité avec “a + (b + c)” et “(a + b) + c”
   - la distributivité avec "(a+b)*c" et “a$c$ + $b$c”

   Pour la commutativité on a :
   a+b $\iff$ a, b, +
   b+a $\iff$ b, a, +
   Pour l’associativité on a :
   a + (b + c) $\iff$ b, c, +, a, +
   (a + b) + c $\iff$ a, b, +, c, +
   Pour la distributivité on a :
   (a + b)$c$ $\iff$ $a$, $b$, +, $c$,
   a$c$ + $b$c $\iff$ a, c, , $b$, $c$, , +

2. Pourquoi une pile est-elle une structure de donnée adaptée pour évaluer une expression en notation polonaise inverse ?

une expression en notation polanaise inverse se lit et s’évalue de gauche à droite en effectuant des opérations successives sur le dernier élément lu.

3. Écrire une fonction qui prend en argument une pile stockant une expression en notation polonaise inverse avec seulement + et * comme opération arithmétique, et qui donne en sortie la valeur de cette expression. Elle prendra par exemple en argument la pile [3,2,+,15,*,6,+] et donne le résultat 81.

[2]:
```
def creer_pile():
    return []

def empiler(p, v):
    p.append(v)

def taille(p):
    return len(p)

def est_vide(p):
    return taille(p) == 0

def depiler(p):
    assert len(p) > 0
    return p.pop()
```

[3]:
```
def evaluer(p):
    p_prime = creer_pile()
    for c in p:
        if c == '+' :
```

```
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x + y)
        elif c == '*' :
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x * y)
        else :
            empiler(p_prime, c)
    resultat = depiler(p_prime)
    return resultat
```

[4]:
```
pile = [3,2,'+',15,'*',6,'+']

resultat = evaluer(pile)

print(resultat)
```

```
81
```

4. Améliorer la fonction précédente pour prendre en compte aussi les opérations - et /

[5]:
```
def evaluer(p):
    p_prime = creer_pile()
    for c in p:
        if c == '+' :
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x + y)
        elif c == '*' :
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x * y)
        elif c == '-' :
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x - y)
        elif c == '/' :
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x / y)
        else :
            empiler(p_prime, c)
    resultat = depiler(p_prime)
    return resultat
```

[6]:
```
pile = [3,2,'+',15,'*',6,'-']
```

```
resultat = evaluer(pile)

print(resultat)
```

```
69
```

5. Enfin améliorer à nouveau la fonction précédente pour prendre en compte l'opération $\sqrt{}$.

Vous pourriez maintenant écrire le logiciel complet pour toutes les touches de la calculatrice en photo, en début d'exercice.

[7]:
```
def evaluer(p):
    p_prime = creer_pile()
    for c in p:
        if c == '+' :
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x + y)
        elif c == '*' :
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x * y)
        elif c == '-' :
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x - y)
        elif c == '/' :
            y = depiler(p_prime)
            x = depiler(p_prime)
            empiler(p_prime, x / y)
        elif c == 'sqrt' :
            x = depiler(p_prime)
            empiler(p_prime, x**0.5)
        else :
            empiler(p_prime, c)
    resultat = depiler(p_prime)
    return resultat
```

[8]:
```
pile = [3,2,'+',15,'*',6*6,'sqrt','-']

resultat = evaluer(pile)

print(resultat)
```

```
69.0
```