

## Correction du TP n°1

### Exercice 1 :

```
Command Window
>> 3+8

ans =

    11
```

(1.1) Le résultat est le suivant :

```
Command Window
>> tan((4*pi)/7)

ans =



-4.381286267534825
```

(1.2) La valeur égale à  $= 2.324205432963649 \cdot 10^2$  : (en Longue Format avec précision de 15 chiffres)

```
Command Window
>> 7^(2.8)

ans =

2.324205432963649e+02
```

(1.3) Lorsqu'on appuie sur  le logiciel matlab nous affiche dans la zone de commande l'ensemble des instructions de commande qui ont été exécutées en avant, tandis que le bouton  permet d'afficher les instructions exécutées en ultérieure.

### Exercice 2 :

(2.1) Le calcul de racine de ln de 6 via une variable est fait comme le suivant :

```
Command Window
>> x= log(6)

x =

    1.791759469228055

>> sqrt(x)

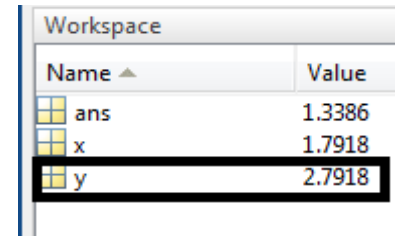
ans =

    1.338566199045850
```

(2.2) La variable y vaut :

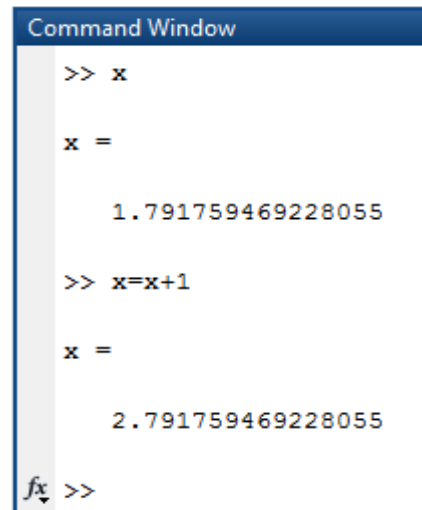
```
>> y=x+1  
  
y =  
  
2.791759469228055
```

Et elle existe bien  
dans l'espace du  
travail



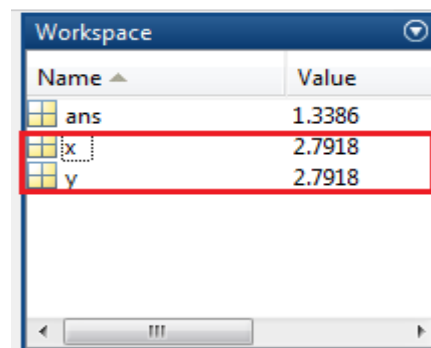
Name	Value
ans	1.3386
x	1.7918
y	2.7918

(2.3) La commande est comme suit :



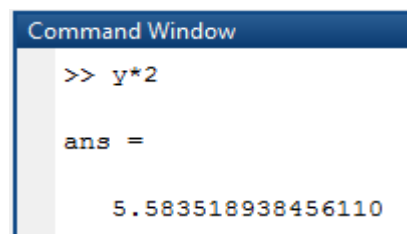
```
>> x  
  
x =  
  
1.791759469228055  
  
>> x=x+1  
  
x =  
  
2.791759469228055  
  
fx >>
```

(2.4) La variable x a maintenant la même valeur que y



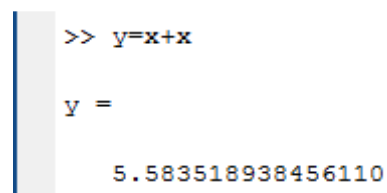
Name	Value
ans	1.3386
x	2.7918
y	2.7918

(2.5) Multiplication de la variable y par 2 :



```
>> y*2  
  
ans =  
  
5.583518938456110
```

(2.6) Pour l'avoir on fait :



```
>> y=x+x  
  
y =  
  
5.583518938456110
```

### Exercice 3 :

(3.1) L'évaluation des quantités est comme le suivant : Premièrement il faut taper **>> format short** pour avoir

5 chiffres significatifs

```
Command Window
>> format short
```

- |   |  |   |
|---|--|---|
| <p>1. <pre>&gt;&gt; tanh(exp(1))</pre></p> <pre>ans =</pre> <p>0.9913</p>   | <p>4. <pre>&gt;&gt; gcd(48972, 36533112)</pre></p> <pre>ans =</pre> <p>48972</p> | <p>7. <pre>&gt;&gt; lcm(318, 732)</pre></p> <pre>ans =</pre> <p>38796</p> |
| <p>2. <pre>&gt;&gt; log10(2)</pre></p> <pre>ans =</pre> <p>0.3010</p>   | <p>5. <pre>&gt;&gt; exp(10^(2.7))</pre></p> <pre>ans =</pre> <p>4.6010e+217</p>  | <p>8. <pre>&gt;&gt; sqrt(3*pi)</pre></p> <pre>ans =</pre> <p>3.0700</p>   |
| <p>3. <pre>&gt;&gt; 1/(1+1/(1+(1/2)))</pre></p> <pre>ans =</pre> <p>0.6000</p>  | <p>6. <pre>&gt;&gt; (cos(pi/4))^(-1)</pre></p> <pre>ans =</pre> <p>1.4142</p>    |   |
| <p>9. <pre>&gt;&gt; log(1/pi)</pre></p> <pre>ans =</pre> <p>-1.1447</p> <p><pre>&gt;&gt; -log(pi)</pre></p> <pre>ans =</pre> <p>-1.1447</p> |  |   |

(3.2) Le calcul de cette expression est comme le suivant :

- La règle de priorité en Matlab est la suivante :

1. Les parenthèses ().
2. La puissance ^ s'applique sur toute l'expression.
3. La multiplication \* et la division / ou \
4. L'addition + et la soustraction -

```
>> 8/100-0.5*(4/10)^2
```

```
ans =
```

-1.3878e-17

Alors que

```
>> 8/100-0.5*(4^2/10^2)
```

```
ans =
```

0

(3.3) Pour calculer les 2 nombres jusqu'au 15<sup>ème</sup> chiffres significatifs on doit utiliser : **>>format long**

Il est remarquable que  
 $382801\pi$  est plus  
 grand que  $e^{14}$

```
Command Window
>> format long
>> exp(14)

ans =

    1.202604284164777e+06

>> 382801*pi

ans =

    1.202604809386826e+06
```

(3.4) Pour connaître la meilleure approximation du  $\sqrt{7}$  on calcule toutes les ratios en format long :

```
Command Window
>> sqrt(7)

ans =

    2.645751311064591
```

C'est la meilleure  
 approximation

```
>> 2709/(2^(10))

ans =

    2.645507812500000
```

```
>> 10583/4000

ans =

    2.645750000000000
```

```
>> 2024/765

ans =

    2.645751633986928
```

```
>> 3/8

ans =

    0.375000000000000
```

(3.5) Prenons un exemple de la division à droite désigne 3 divise 8 tandis que

```
>> 3\8

ans =

    2.666666666666667
```

la division à gauche

désigne le contraire.

(3.6) Pour trouver les parties réelles et imaginaires des nombres complexes on fait :

```
>> z=exp(i*((3*pi)+4))

z =

    0.653643620863612 + 0.756802495307928i
```

```
>> real(z)

ans =

    0.653643620863612
```

```
>> imag(z)

ans =

    0.756802495307928
```

On fait la même chose  
 pour l'ensemble des  
 nombres complexes

(3.7) Pour calculer les normes on utilise **abs()** et pour calculer les arguments des nombres complexes on utilise **angle()** :

```
>> z=exp(i*((3*pi)+4))
```

```
z =
```

```
0.653643620863612 + 0.756802495307928i
```

```
>> angle(z)
```

```
ans =
```

```
0.858407346410206
```

```
>> abs(z)
```

```
ans =
```

```
1.0000000000000000
```

On fait la même chose  
pour les autres nombres  
complexes