

Exercices d'applications des mathématiques - Corrigé de la série n° 1

Cours 3AMOS01

1. Gravitation.

On trouve pour g

(1) à la surface de la Terre à l'équateur

```
octave:1> 6.67259*10^-11*5.9742*10^24/(6.378140*10^6)^2  
ans = 9.7991
```

(2) à la surface de la Lune

```
octave:2> 6.67259*10^-11*7.350*10^22/(1.738*10^6)^2  
ans = 1.6236
```

2. La Lune.

On trouve

```
octave:1> 7.350*10^22/(4/3*pi*(1.738*10^6)^3)  
ans = 3342.3
```

3. Variables.

Il suffit de compléter le code comme suit:

```
octave:2> G=6.67259*10^-11  
G = 6.6726e-11  
octave:3> M=7.350*10^22  
M = 7.3500e+22  
octave:4> R=1.737*10^6  
R = 1737000  
octave:5> G*M/R^2  
ans = 1.6255
```

4. Variables.

On trouve

```
octave:1> gustave=2;  
octave:2> gustave=gustave^2;  
octave:3> gustave=gustave^2;  
octave:4> gustave=gustave^2;  
octave:5> gustave=gustave^2;  
octave:6> gustave  
gustave = 65536
```

5. Tableaux.

Il suffit d'exécuter le code *Octave* suivant:

```
octave:1> x=[1:2:99];  
octave:2> x  
x =
```

Columns 1 through 16:

1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

Columns 17 through 32:

33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63

Columns 33 through 48:

65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95

Columns 49 and 50:

97 99

Il suffit de taper:

```
octave:3> x.^2
ans =
```

Columns 1 through 11:

1 9 25 49 81 121 169 225 289 361 441

Columns 12 through 22:

529 625 729 841 961 1089 1225 1369 1521 1681 1849

Columns 23 through 33:

2025 2209 2401 2601 2809 3025 3249 3481 3721 3969 4225

Columns 34 through 44:

4489 4761 5041 5329 5625 5929 6241 6561 6889 7225 7569

Columns 45 through 50:

7921 8281 8649 9025 9409 9801

6. Boucles.

En sauvegardant sous le nom boucle.m et en exécutant le script suivant:

```
1 for n=1:1:10
2   printf('Bonjour !\n')
3 end
```

on obtient:

```
octave:1> boucle
Bonjour !
Bonjour !
Bonjour !
Bonjour !
Bonjour !
```

```

Bonjour !
Bonjour !
Bonjour !
Bonjour !
Bonjour !
octave:2>

```

7. Boucles.

En exécutant le script suivant:

```

1 for n=1:2:99
2   n^2
3 end

```

on obtient les carrés de tous les nombres impairs entre 1 et 99.

8. Boucles.

On peut, par exemple, exécuter le script suivant:

```

1 for n=1:2:26
2   char(n+96)
3 end

```

9. Boucles.

On peut, par exemple, utiliser le script suivant:

```

1 texte='bonjour';
2 l=length(texte);
3 for n=1:l
4   printf(char(toascii(texte(n))-32))
5 end
6 printf('\n')

```

10. $\text{mod}(a,b)$.

La commande `mod(a,b)` renvoie le nombre $0 \leq p < b$ congru à a modulo b .

11. $\text{mod}(a,b)$.

La commande `mod([0:1:100],26)` renvoie un tableau de 101 cases dont la case i contient `mod(i,26)`.

12. Problème.

L'exécution du code

```

1 n=[-100:1:100];
2 plot(n,mod(n,26),'@1')

```

dont le graphique de la figure 1

13. `toascii` et `char`.

La commande `toascii('bonjourcommentallezvous')` donne

```

octave:11> toascii('bonjourcommentallezvous')
ans =

    98   111   110   106   111   117   114   99   111   109   109   101
  110   116   97   108   108   101   122   118   111   117   115

```

C'est un tableau dont les cases contiennent les code ASCII des lettres 'bonjourcommentallezvous'.

Par ailleurs, la commande `char([98,105,101,110])` donne

```

octave:12> char([98,105,101,110])
ans = bien

```

C'est une chaîne de caractères dont les codes ASCII sont les nombres 98, 105, 101 et 110.

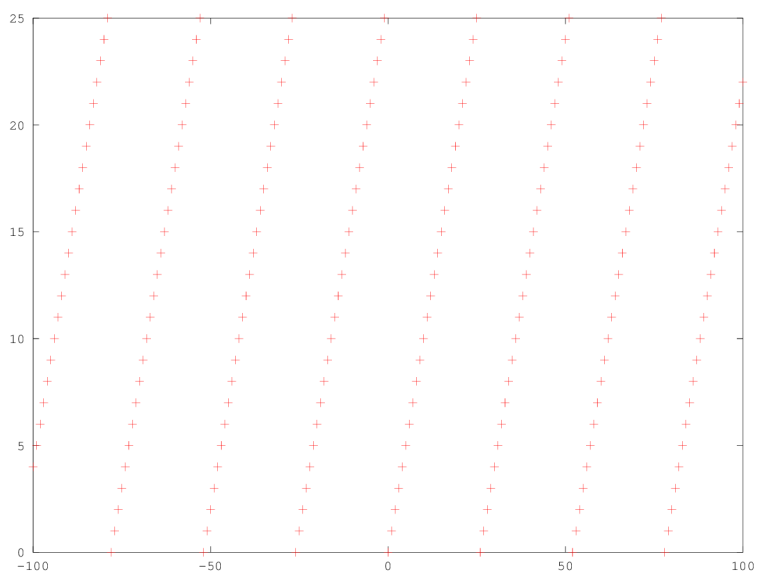


FIGURE 1. Exercice 12