

Examen final (Calcul formel et programmation)**Exercice 1.(5 pt)**

a- Donner le résultat de chacune des instructions Maple suivantes:

- 1) $> \text{diff}(x^3 + 2*x, x);$ 2) $> \text{diff}(x^3 + 2*x, x^2);$
3) $> 9 \bmod 4;$ 4) $> \text{expand}(\cos(a+b));$

b- Traduire les expressions mathématiques suivantes en instructions Maple

- 1) $\text{pgcd}(a,b)$ 2) $\text{ppcm}(a,b)$

c- Corriger les erreurs de syntaxe dans les commandes suivantes :

- 1) $> \text{With}(\text{Plots});$
 $> \text{Plots}([2.\cos(t)+3.\sin(t), t=0 \dots 2*\pi], \text{color}="blue");$
2) $> \text{With}(\text{plots});$
 $> \text{plot3D}(x^2 + y^2=0, x = -1..1, y = -1..1);$

Exercice 2.(5 pt)

On considère la matrice $A = \begin{pmatrix} 2 & 1 & -1 \\ 1 & 0 & 1 \\ -1 & 1 & 2 \end{pmatrix}$

En utilisant les instructions de Maple seulement

- 1) Diagonaliser la matrice A grâce à la commande JordanForm .
Vérifier que $A = PDP^{-1}$;
- 2) Calculer A multiplier par sa transposée ;
- 3) Calculer et factoriser le polynôme caractéristique de A ;
- 4) Résoudre le système $AX=b$ pour $b = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$.

Exercice 3 .(5 pt)

- a) 1) Dessiner la surface S_1 d'équation $z = x^2 + y^2$ pour $-2 < x < 2$ et $-2 < y < 2$, en couleur bleue.
- 2) Dessiner la surface S_2 d'équation $z = 4 - x^2 - y^2$ pour $-2 < x < 2$ et $-2 < y < 2$, en couleur rouge.
- 3) Tracer l'intersection des surfaces S_1 et S_2 .
- b) 1) Ecrire une procédure qui prend en entrée une fonction et qui trace la fonction et sa dérivée seconde en sortie.
- 2) Tester la procédure en traçant la fonction cos.

Exercice 4.(5pt)

- a) 1) Exécuter les instructions suivantes :
- ```
> with(numtheory) :
> k:=1: while tau(k)< >20 do k:=k+1 od: print(k);
> divisors(240);
> nops(divisors(240));
```
- 2) Trouver le plus petit nombre ayant exactement 15 diviseurs.
- b) 1) La procédure suivante contient des erreurs de syntaxe. Les corriger.
- ```
pd := proc(a, b) ;  
if b = 0 else  
    return(a);  
then  
    return(pd(b, irem(a, b)));  
fi;  
end;
```
- 2) Tester la procédure pour le couple (a,b)=(12345, 98765).

c) On définit la suite de Fibonacci comme suit

$$\begin{cases} u_0 = 1 \\ u_1 = 1 \\ u_{n+2} = u_{n+1} + u_n \text{ pour } n \geq 0 \end{cases}$$

- 1) Ecrire une procédure récursive **Fibo_rec** qui prend en entrée un entier n et qui donne le nième terme de la suite de Fibonacci.
- 2) calculer u_{30} .

Corrigé type de l'examen final

```
> #Exercice 1 [5]
> #a) [0.5]x4
> diff(x^3 + 2*x, x); diff(x^3 + 2*x, x$2);
```

$$3x^2 + 2$$

$$6x$$

```
> 9 mod 4; expand(cos(a+b));
```

$$1$$

$$\cos(a)\cos(b) - \sin(a)\sin(b)$$

```
> #b) [1]
> [0.5]+[0.5]
#1)>igcd(a,b) 2)>ilcm(a,b)
> #c) [2]
> [1]+[1]
> #1) with(plots):
> #2) plot([2*cos(t)+3*sin(t),,t=0...2*Pi],color=blue);
> #2) with(plots):
```

```
> plot3d(x^2 + y^2, x = -1..1, y = -1..1);
> #Exercice 02 [5] ([2]+[1]+[1]+[1])
> restart;
> with(LinearAlgebra):
> A:=<<2 ,1,-1>| <1,0,1>| <-1,1,2>>>;
```

$$A := \begin{bmatrix} 2 & 1 & -1 \\ 1 & 0 & 1 \\ -1 & 1 & 2 \end{bmatrix}$$

```
> Delta:=JordanForm(A);
```

$$\Delta := \begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

```
> P:=JordanForm(A,output='Q');
```

$$P := \begin{bmatrix} \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \\ -\frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{6} & \frac{1}{3} & -\frac{1}{2} \end{bmatrix}$$

```
> A:=P.Delta.P^(-1);
```

$$A := \begin{bmatrix} 2 & 1 & -1 \\ 1 & 0 & 1 \\ -1 & 1 & 2 \end{bmatrix}$$

```
> A.Transpose(A);
```

$$\begin{bmatrix} 6 & 1 & -3 \\ 1 & 2 & 1 \\ -3 & 1 & 6 \end{bmatrix}$$

```
> CharacteristicPolynomial(A,x);
```

$$x^3 - 4x^2 + x + 6$$

```
> factor(%);
```

$$(x - 2)(x - 3)(x + 1)$$

```
> b:=<0,1,0>;
```

$$b := \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

```
> x:=LinearSolve(A,b,free='lambda');
```

$$x := \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

```
> restart;
```

```
> #Exercice 03 [5] ( [3]+[2])
```

```
> with(plots):
```

Warning, the name changecoords has been redefined

```
> S1 := plot3d(x^2 + y^2, x = -2..2, y = -2..2, color = blue):
```

```
> S2 := plot3d(4 - x^2 - y^2, x = -2..2, y = -2..2,color = red):
```

```
> display([S1, S2]);
```

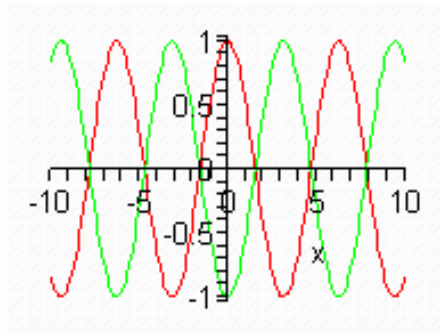


```
> restart;

> #b)
> with (plots):
>   dessin:=proc(f)
local g;
g:=diff(f(x),x$2);
plot([f(x),g(x)],x);
end;

dessin := proc(f) local g; g := diff(f(x), `$(x, 2)); plot([f(x), g(x)], x); end proc;

> dessin(cos);
```



```
> #Exercise 4 [5]
> #a [1.5]
> restart:
> with(numtheory) :
k:=1: while tau(k)<>20 do k:=k+1 od: print(k);
```

240

```
> divisors(240);nops(divisors(240));
{1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 40, 48, 60, 80, 120, 240}
```

20

```
> #2
> k:=1: while tau(k)<>15 do k:=k+1 od: print(k);
```

144

```
> divisors(144);nops(divisors(144));
{1, 2, 3, 4, 6, 8, 9, 12, 16, 18, 24, 36, 48, 72, 144}
```

15

```

> #b) [1.5]
> restart;
> pd:=proc(a,b)
  if b=0 then
    return(a);
  else
    return(pd(b,irem(a,b)));
  fi;
end;
  pd := proc(a, b) if b = 0 then return a else return pd(b, irem(a, b)) end if; end proc;
> pd(12345, 98765);

```

5

```

> #c) [2]
> Fibo_rec := proc(n)
  if n = 0 then
    return 1;
  elif n = 1 then
    return 1;
  else
    return Fibo_rec(n-1) + Fibo_rec(n-2);
  end if;
end proc :
> Fibo_rec(30);

```

1346269

>