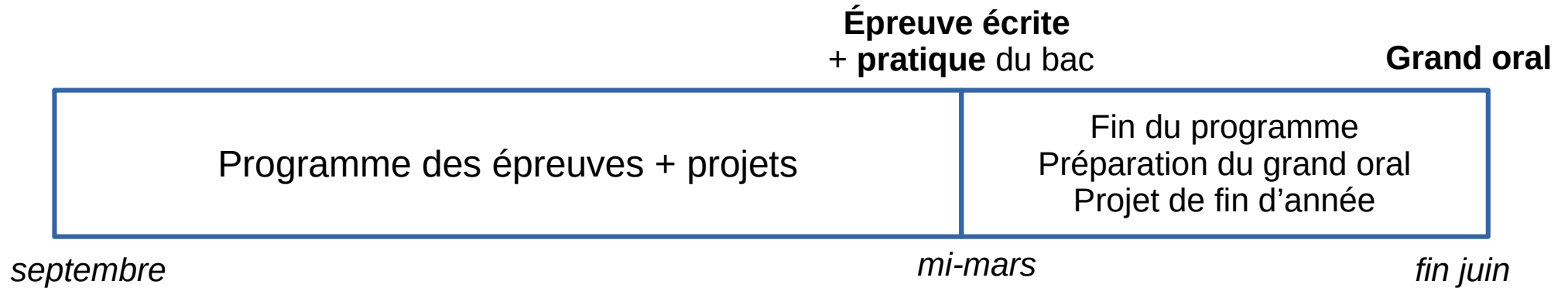


# Programme de terminale NSI

L. TREBAUL

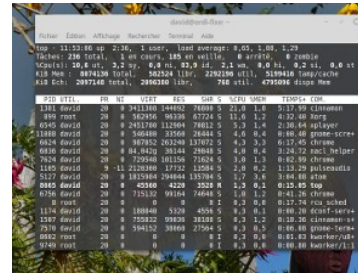
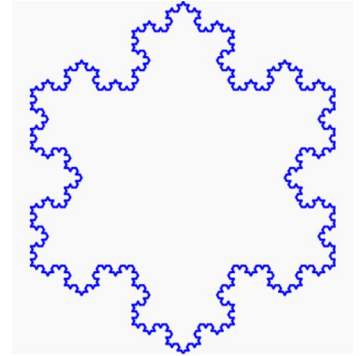
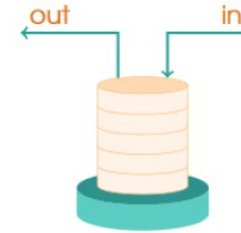
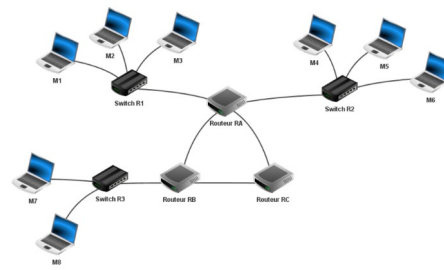
# Organisation

- 6h par semaine
- Une année rythmée par le bac :



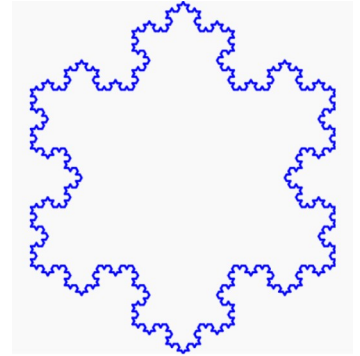
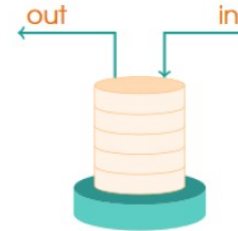
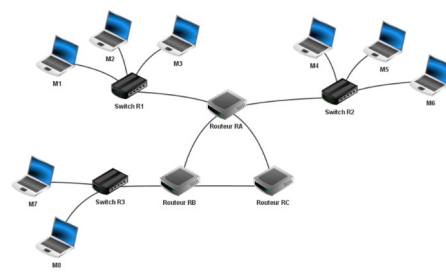
# Programme

- 1) Programmation
- 2) Structures de données
- 3) Réseaux informatiques
- 4) Architecture matérielle & systèmes d'exploitation
- 5) Bases de données
- 6) Algorithmique



# Programme

- 1) Programmation
- 2) Structures de données
- 3) Réseaux informatiques
- 4) Architecture matérielle & systèmes d'exploitation
- 5) Bases de données
- 6) Algorithmique

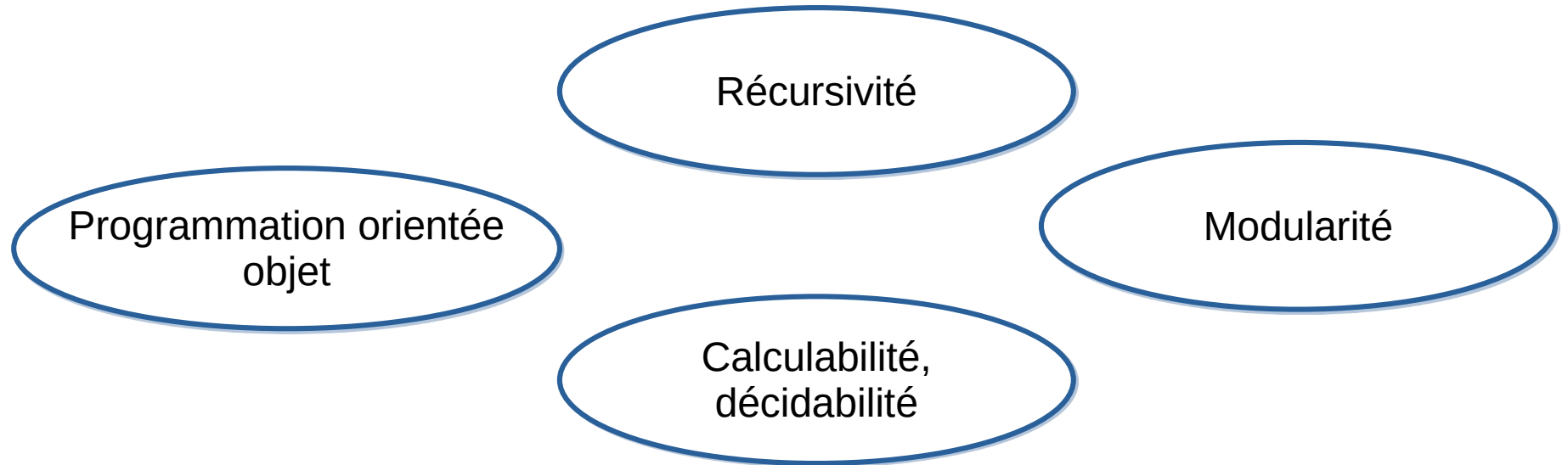


A screenshot of a terminal window displaying system statistics and a table of process data. The top section shows system-wide metrics like CPU usage, memory usage, and disk I/O. Below this is a table with columns for PID, USER, PPID, CPU, MEM, and COMMAND, listing various running processes.

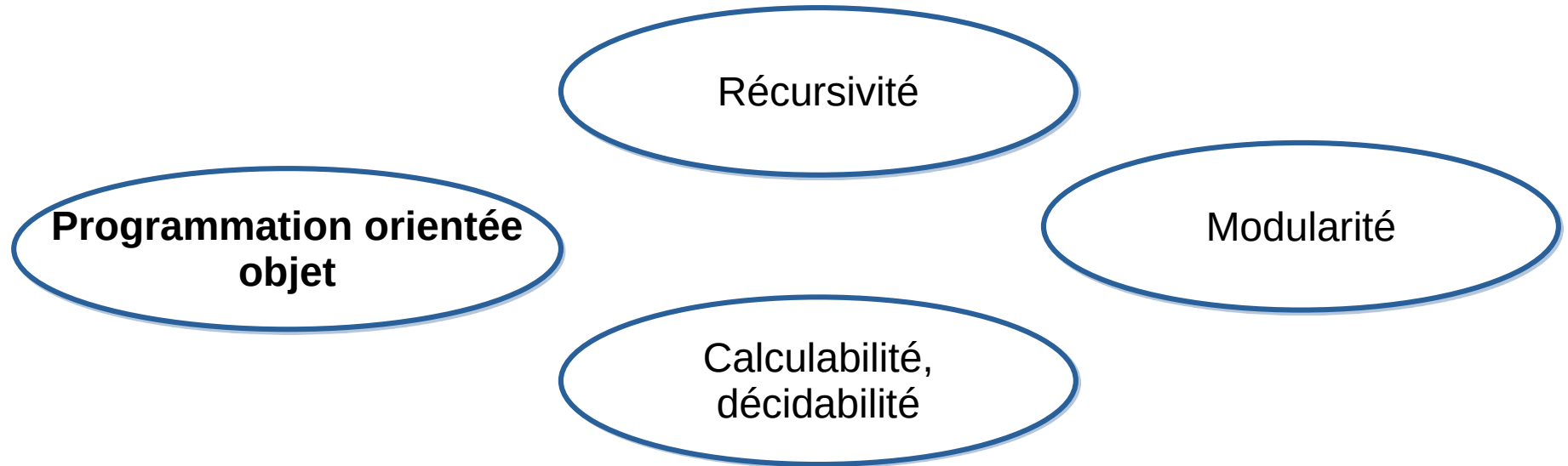
PID	USER	PPID	CPU	MEM	COMMAND
1	root	0	0.0	0.0	init
2	root	0	0.0	0.0	smd
3	root	0	0.0	0.0	smd
4	root	0	0.0	0.0	smd
5	root	0	0.0	0.0	smd
6	root	0	0.0	0.0	smd
7	root	0	0.0	0.0	smd
8	root	0	0.0	0.0	smd
9	root	0	0.0	0.0	smd
10	root	0	0.0	0.0	smd
11	root	0	0.0	0.0	smd
12	root	0	0.0	0.0	smd
13	root	0	0.0	0.0	smd
14	root	0	0.0	0.0	smd
15	root	0	0.0	0.0	smd
16	root	0	0.0	0.0	smd
17	root	0	0.0	0.0	smd
18	root	0	0.0	0.0	smd
19	root	0	0.0	0.0	smd
20	root	0	0.0	0.0	smd
21	root	0	0.0	0.0	smd
22	root	0	0.0	0.0	smd
23	root	0	0.0	0.0	smd
24	root	0	0.0	0.0	smd
25	root	0	0.0	0.0	smd
26	root	0	0.0	0.0	smd
27	root	0	0.0	0.0	smd
28	root	0	0.0	0.0	smd
29	root	0	0.0	0.0	smd
30	root	0	0.0	0.0	smd
31	root	0	0.0	0.0	smd
32	root	0	0.0	0.0	smd
33	root	0	0.0	0.0	smd
34	root	0	0.0	0.0	smd
35	root	0	0.0	0.0	smd
36	root	0	0.0	0.0	smd
37	root	0	0.0	0.0	smd
38	root	0	0.0	0.0	smd
39	root	0	0.0	0.0	smd
40	root	0	0.0	0.0	smd
41	root	0	0.0	0.0	smd
42	root	0	0.0	0.0	smd
43	root	0	0.0	0.0	smd
44	root	0	0.0	0.0	smd
45	root	0	0.0	0.0	smd
46	root	0	0.0	0.0	smd
47	root	0	0.0	0.0	smd
48	root	0	0.0	0.0	smd
49	root	0	0.0	0.0	smd
50	root	0	0.0	0.0	smd
51	root	0	0.0	0.0	smd
52	root	0	0.0	0.0	smd
53	root	0	0.0	0.0	smd
54	root	0	0.0	0.0	smd
55	root	0	0.0	0.0	smd
56	root	0	0.0	0.0	smd
57	root	0	0.0	0.0	smd
58	root	0	0.0	0.0	smd
59	root	0	0.0	0.0	smd
60	root	0	0.0	0.0	smd
61	root	0	0.0	0.0	smd
62	root	0	0.0	0.0	smd
63	root	0	0.0	0.0	smd
64	root	0	0.0	0.0	smd
65	root	0	0.0	0.0	smd
66	root	0	0.0	0.0	smd
67	root	0	0.0	0.0	smd
68	root	0	0.0	0.0	smd
69	root	0	0.0	0.0	smd
70	root	0	0.0	0.0	smd
71	root	0	0.0	0.0	smd
72	root	0	0.0	0.0	smd
73	root	0	0.0	0.0	smd
74	root	0	0.0	0.0	smd
75	root	0	0.0	0.0	smd
76	root	0	0.0	0.0	smd
77	root	0	0.0	0.0	smd
78	root	0	0.0	0.0	smd
79	root	0	0.0	0.0	smd
80	root	0	0.0	0.0	smd
81	root	0	0.0	0.0	smd
82	root	0	0.0	0.0	smd
83	root	0	0.0	0.0	smd
84	root	0	0.0	0.0	smd
85	root	0	0.0	0.0	smd
86	root	0	0.0	0.0	smd
87	root	0	0.0	0.0	smd
88	root	0	0.0	0.0	smd
89	root	0	0.0	0.0	smd
90	root	0	0.0	0.0	smd
91	root	0	0.0	0.0	smd
92	root	0	0.0	0.0	smd
93	root	0	0.0	0.0	smd
94	root	0	0.0	0.0	smd
95	root	0	0.0	0.0	smd
96	root	0	0.0	0.0	smd
97	root	0	0.0	0.0	smd
98	root	0	0.0	0.0	smd
99	root	0	0.0	0.0	smd
100	root	0	0.0	0.0	smd



# 1- Programmation

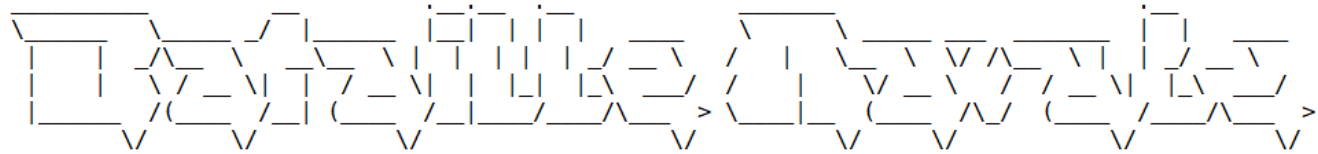


# 1- Programmation



# Projets en Programmation Orientée Objet

## Un jeu de bataille navale



```
print("Placement du torpilleur (deux cases)")
while torpilleurUser.setPlace(x, y, direction) == False:
    while True:
        try:
            x = ligne[input("Entrez une ligne (entre A et J): ")]
            break
        except KeyError:
            x = ligne[input("Entrez une ligne (entre A et J): ")]
    while True:
        try:
            y = int(input("Entrez une colonne (Entre 1 et 10): "))
            break
        except ValueError:
            y = int(input("Entrez une colonne (Entre 1 et 10): "))
    while True:
        try:
            direction = orientation[input("Entrez une direction (haut, droite, gauche, bas): ")]
            break
        except KeyError:
            direction = orientation[input("Entrez une direction (haut, droite, gauche, bas): ")]
    torpilleurUser.setPlace(x, y, direction)
print("Les coordonnées que vous avez entré ne fonctionnent pas, veuillez réessayer")

print(tab(l))

x = 0
y = 0
direction = 0

sousmarinUser.setPlace(x, y, direction)

print("Placement du sous-marin (trois cases)")
while sousmarinUser.setPlace(x, y, direction) == False:
    while True:
        try:
            x = ligne[input("Entrez une ligne (entre A et J): ")]
            break
        except KeyError:
            x = ligne[input("Entrez une ligne (entre A et J): ")]
    while True:
```

```
[' ', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
['A', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
['B', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
['C', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
['D', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
['E', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
['F', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
['G', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
['H', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
['I', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
['J', '~', '~', '~', '~', '~', '~', '~', '~', '~', '~']
```

Placement du torpilleur (deux cases)

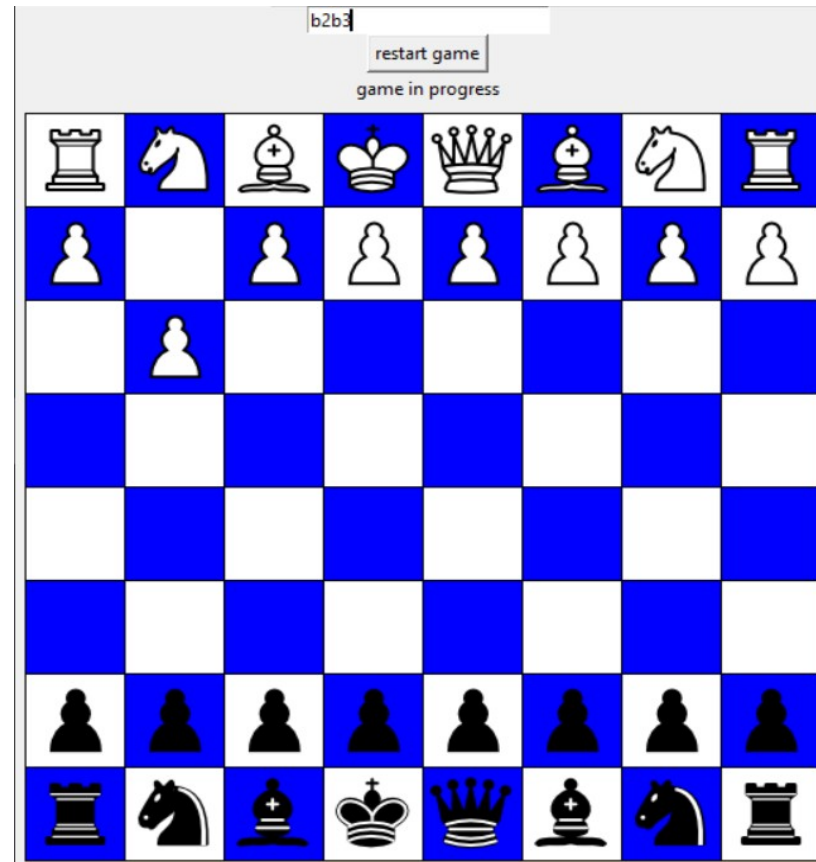
Entrez une ligne (entre A et J):



# Projets en Programmation Orientée Objet

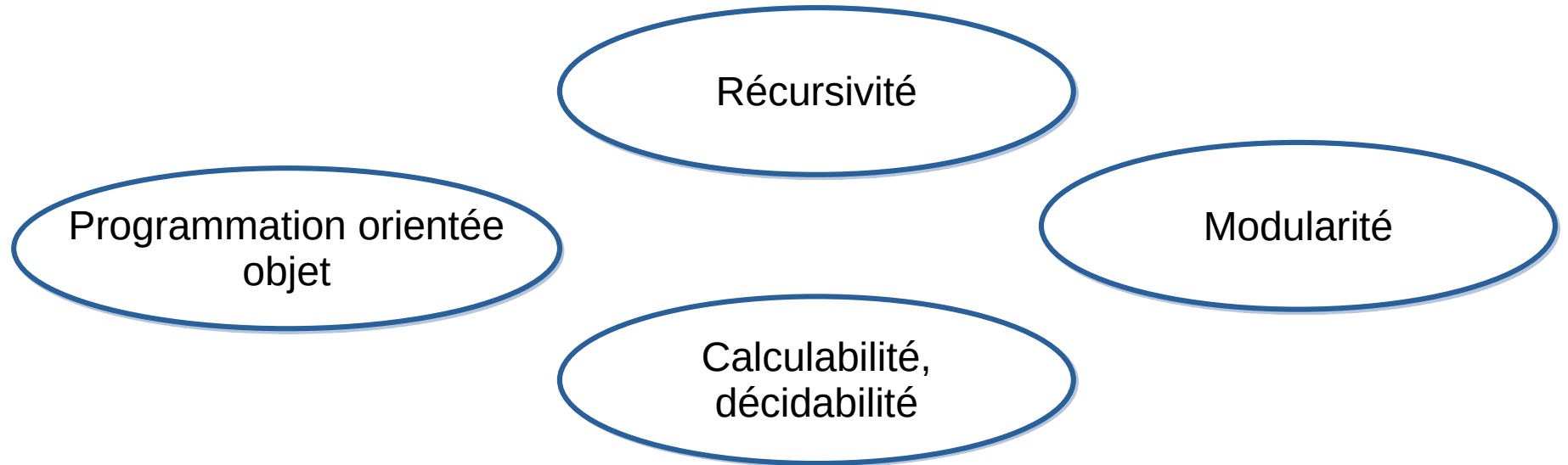
## Un jeu d'échecs : vérification par l'ordinateur de la validité des coups

```
class Pawn(Piece):  
  
    def basepos(self): #on établit la position initiale du pion pour savoir de combien i  
        if self.color=="b":  
            return 6  
        else: return 1  
  
    def __init__(self, color, piece, y, x): #init, on prend en considération sa position e  
        super().__init__(color, piece, y, x)  
        if self.color == "b":  
            self.advancement = -1  
        else:  
            self.advancement = 1  
        self.value = 1  
  
    def getmoverange(self): #moverange du pion, dépend de multiples facteurs, renvoie un  
        moverange = []  
        if isempty(board, self.y+self.advancement, self.x): #avancer de 1  
            moverange.append((self.y+self.advancement, self.x))  
        if self.y == self.basepos():  
            if isempty(board, self.y+2*self.advancement, self.x) and isempty(board, self.y+  
                moverange.append((self.y+2*self.advancement, self.x))  
        if self.x<7:  
            if not isempty(board, self.y+self.advancement, self.x+1):  
                if self.color != checkboard(board, self.y+self.advancement, self.x+1):  
                    moverange.append((self.y+self.advancement, self.x+1))  
        if 0<self.x:  
            if not isempty(board, self.y+self.advancement, self.x-1):  
                if self.color != checkboard(board, self.y+self.advancement, self.x-1):  
                    moverange.append((self.y+self.advancement, self.x-1))  
        return moverange  
  
    def geteatrange(self): #les endroits où le pion peut manger une pièce  
        eatrange = []  
        if self.x<7:  
            if self.color != checkboard(board, self.y+self.advancement, self.x+1):  
                eatrange.append((self.y+self.advancement, self.x+1))  
        if 0<self.x:  
            if self.color != checkboard(board, self.y+self.advancement, self.x-1):  
                eatrange.append((self.y+self.advancement, self.x-1))  
        return eatrange
```





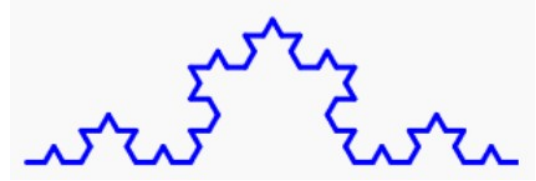
# 1- Programmation



# 1- Programmation



Les tours de Hanoï



Le flocon de von Koch

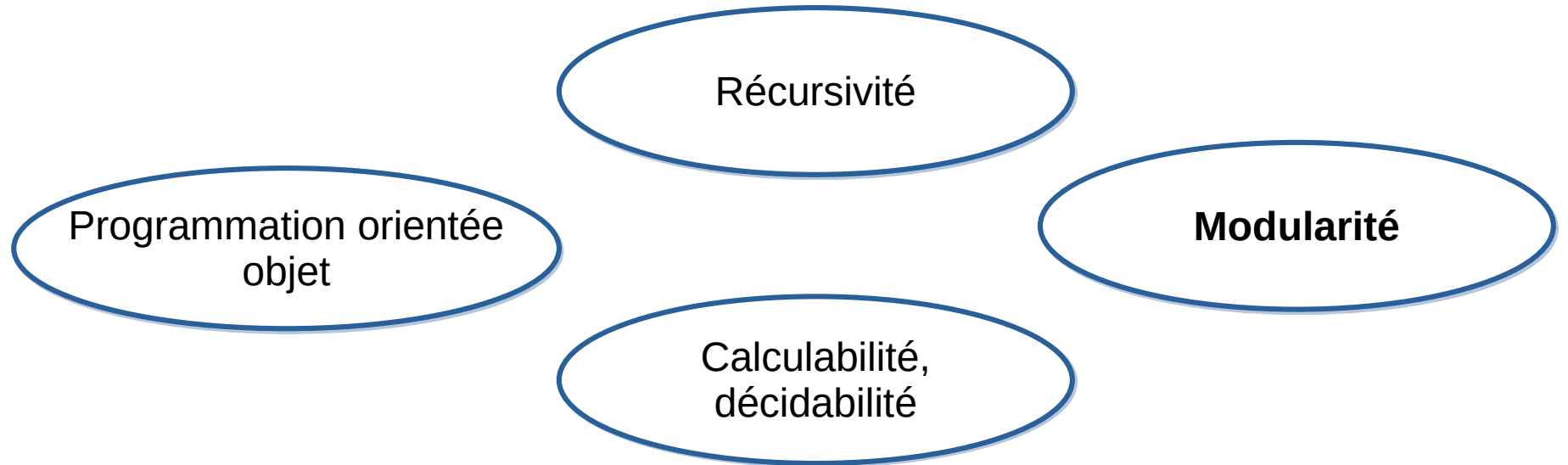
**Récurtivité**

Programmation orientée  
objet

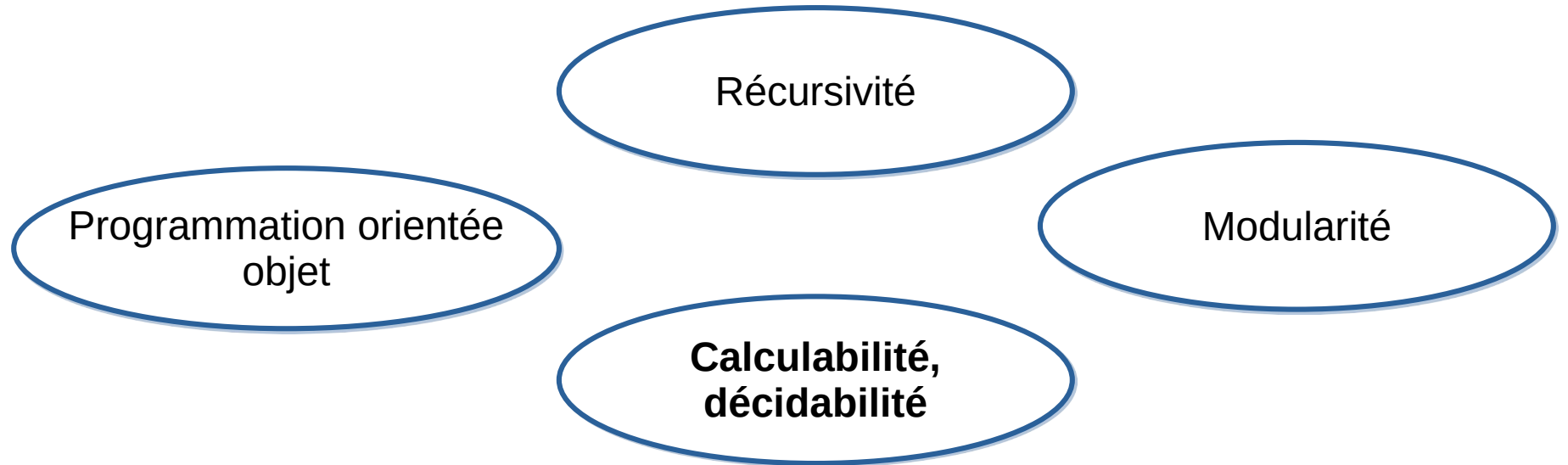
Modularité

Calculabilité,  
décidabilité

# 1- Programmation

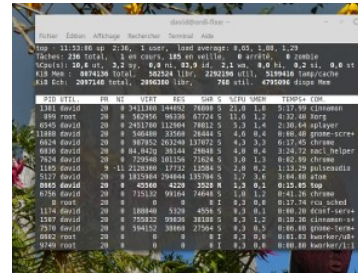
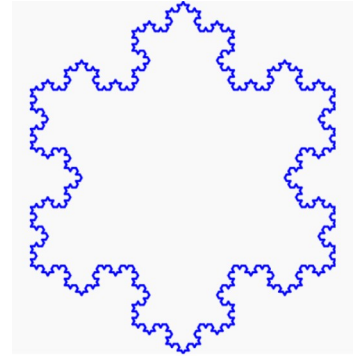
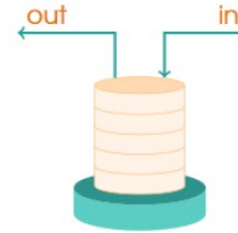
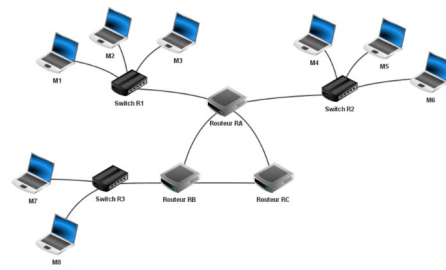


# 1- Programmation



# Programme

- 1) Programmation
- 2) Structures de données
- 3) Réseaux informatiques
- 4) Architecture matérielle & systèmes d'exploitation
- 5) Bases de données
- 6) Algorithmique

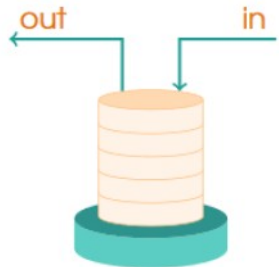


## 2- Structures de données

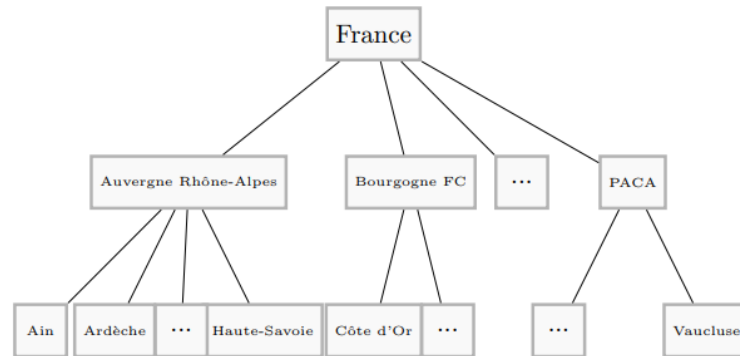
Comment agencer les données  
pour mieux répondre à des problèmes ?



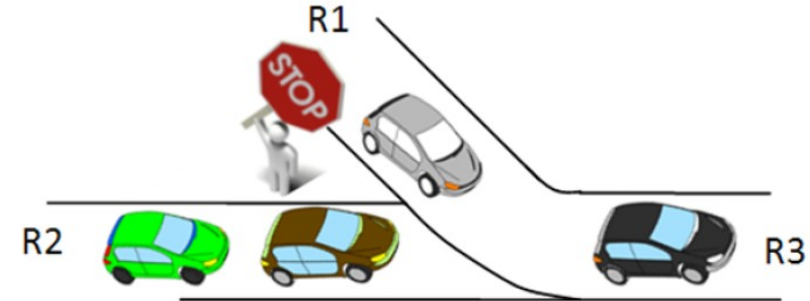
Liste chaînée



Pile



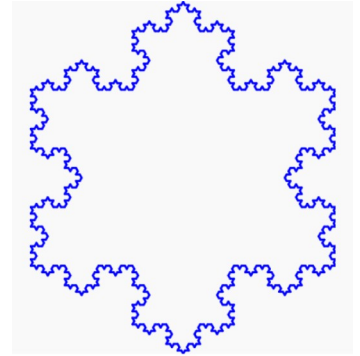
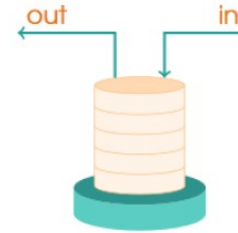
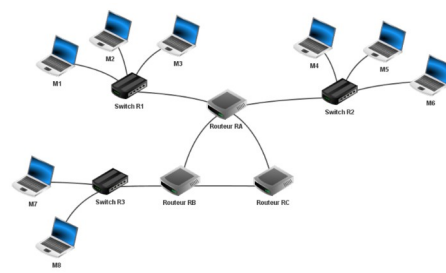
Arbre



Utilisation des  
files

# Programme

- 1) Programmation
- 2) Structures de données
- 3) Réseaux informatiques**
- 4) Architecture matérielle & systèmes d'exploitation
- 5) Bases de données
- 6) Algorithmique



```
top - 11:32:00 up 2:46, 1 user, load average: 0.03, 1.08, 1.28
Device: 280 MB/s, 1.00 GB, 1.00 GB, 0.00 GB, 0.00 GB, 0.00 GB
VCPU(s): 10.0 ut, 5.2 st, 6.0 nt, 82.9 id, 2.1 wa, 0.0 hi, 0.2 si, 0.0 st
KIB Mem: 1874128 Total, 581224 Free, 2392208 Util, 320848 Inactive
KIB Bch: 2097248 total, 2096260 lib, 740 util, 4795086 disp Mem
```

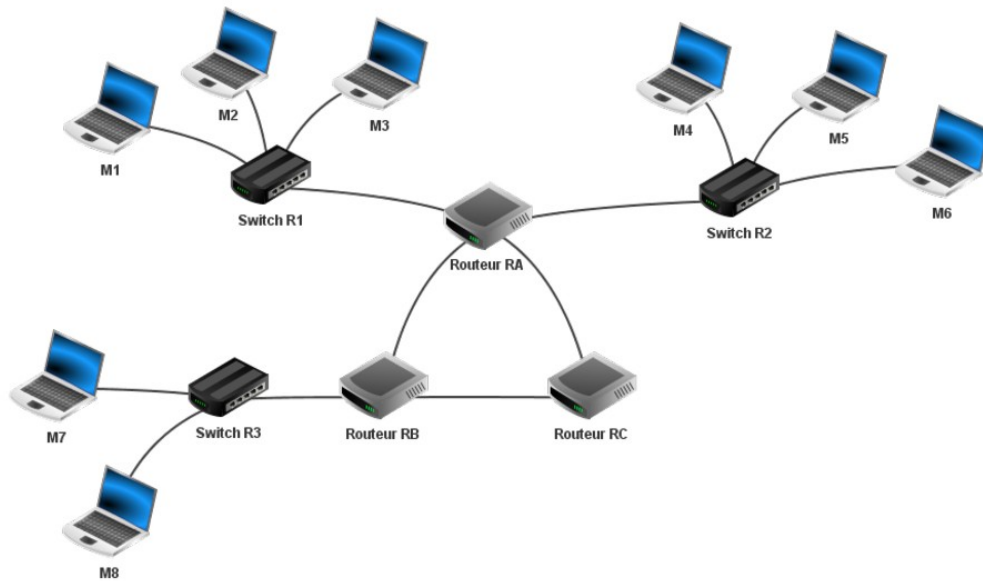
PID	UTL	PP	NI	UPT	RES	GRP	S	CPUS	MEM	VSZ	TIME	COM
1381	0.00	0	0	0	14480	74000	S	21.0	1.0	5.17	90	climacore
888	0.00	0	0	0	50250	94330	S	11.0	1.2	4.52	40	borg
4245	0.00	0	0	0	2451700	112504	S	5.3	1.4	2.38	84	qplayer
1380	0.00	0	0	0	150400	17564	S	4.6	0.4	0.86	48	gnome-scre
6624	0.00	0	0	0	987832	163340	S	4.3	3.3	6.17	45	chrome
8105	0.00	0	0	0	84.0420	25144	S	4.0	0.4	2.42	12	gnome-bell
7624	0.00	0	0	0	729540	181150	S	3.8	1.3	0.82	89	chrome
1120	0.00	0	0	0	2220800	12732	S	2.8	0.2	3.12	20	polymodel
5127	0.00	0	0	0	1815884	294044	S	1.7	3.4	3.44	88	atlas
4882	0.00	0	0	0	45588	4428	S	1.0	0.1	0.15	45	top
8756	0.00	0	0	0	155320	99164	S	1.0	1.2	0.61	35	chrome
0	0.00	0	0	0	0	0	S	0.0	0.0	0.17	74	gnome-sch
1174	0.00	0	0	0	1089400	5120	S	0.5	0.1	0.86	20	gnome-sch
1287	0.00	0	0	0	715822	93830	S	0.3	1.2	0.18	30	climacore++
2716	0.00	0	0	0	194120	1860	S	0.3	0.3	0.86	48	gnome-bell
8882	0.00	0	0	0	0	0	S	0.0	0.0	0.41	83	gnome-bell
1740	0.00	0	0	0	0	0	S	0.0	0.0	0.86	88	gnome-bell





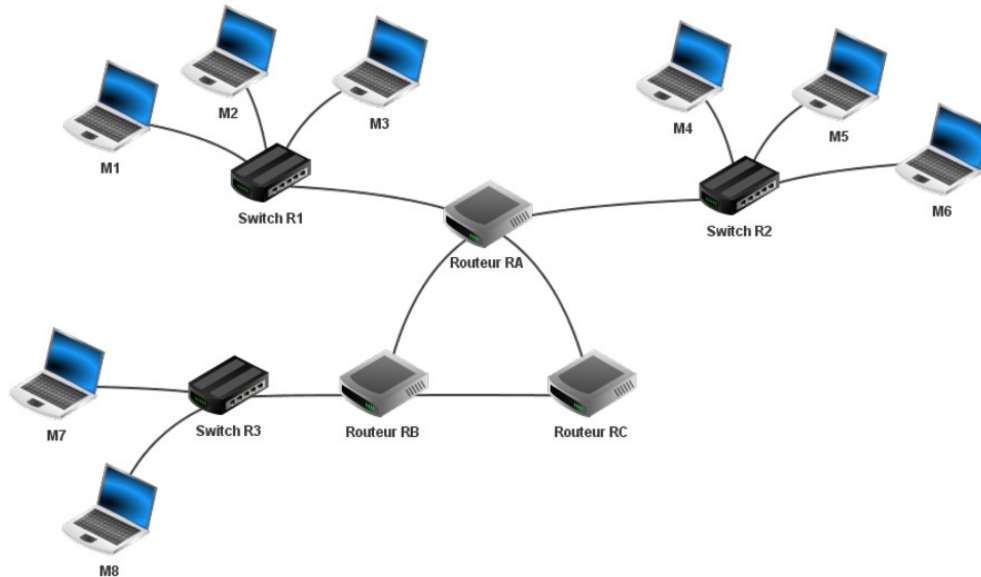
### 3- Réseaux informatiques

Comment *guider* les paquets  
dans un réseau ?

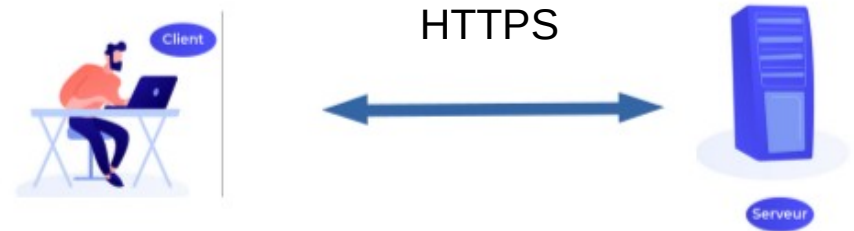


# 3- Réseaux informatiques

Comment *guider* les paquets dans un réseau ?

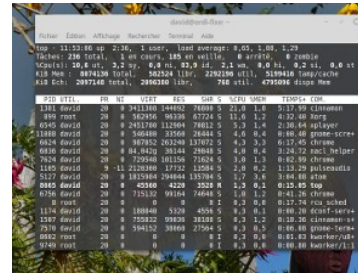
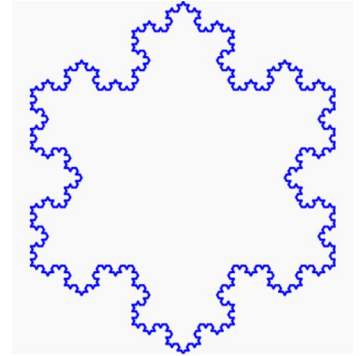
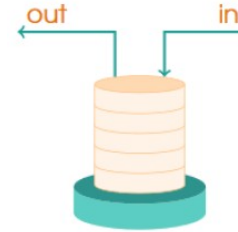
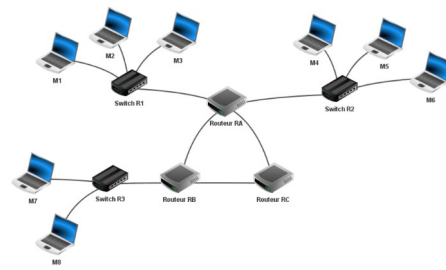


Comment *sécuriser* l'échange d'informations ?



# Programme

- 1) Programmation
- 2) Structures de données
- 3) Réseaux informatiques
- 4) Architecture matérielle & systèmes d'exploitation
- 5) Bases de données
- 6) Algorithmique



# 4- Architecture matérielle et systèmes d'exploitation



Des ordinateurs...

Gestion des ressources  
par un OS

```
david@ordi-ls: ~  
Fichier Édition Affichage Recherche Terminal Aide  
top - 11:53:06 up 2:36, 1 user, load average: 0.05, 1.03, 1.23  
Tâches: 236 total, 1 en cours, 185 en veille, 0 arrêté, 0 zombie  
%Cpu(s): 10.6 us, 3.2 sy, 0.0 ni, 83.9 id, 2.1 wa, 0.0 hi, 0.2 si, 0.0 st  
Mem: 8874136 total, 582524 libre, 2292196 util, 5159416 tasp/cache  
Mem Emb: 2097140 total, 2096380 libre, 760 util, 4795096 dispo Mem
```

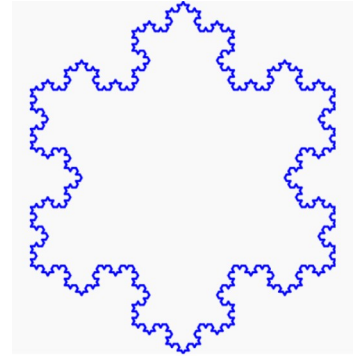
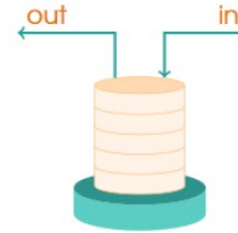
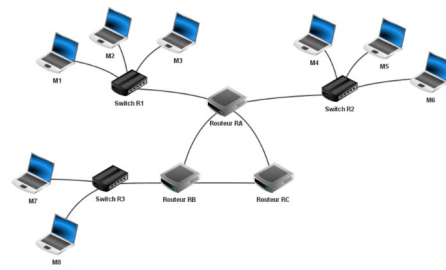
PID	UTIL	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COM.
1361	0.0	20	0	3411360	144692	76096	S	21.0	1.6	5:17.99	cinnamon
899	0.0	20	0	562956	96136	67724	S	11.6	1.2	4:32.48	xorg
6545	0.0	20	0	2451760	112904	78812	S	5.3	1.4	2:38.64	xplayer
11888	0.0	20	0	546488	33568	26444	S	4.6	0.4	8:08.48	gnome-scre+
6624	0.0	20	0	967852	263240	137072	S	4.3	3.3	6:17.45	chrome
6836	0.0	20	0	84.8420	38144	29848	S	4.8	0.4	3:24.72	nacl helper
7624	0.0	20	0	729548	101156	71624	S	3.0	1.3	0:42.99	chrome
1185	0.0	9	-11	2128368	17732	13584	S	2.8	0.2	1:13.29	pulseaudio
5127	0.0	20	0	1815984	294844	135704	S	1.7	3.6	3:04.88	atom
8865	0.0	20	0	45360	4220	3528	N	1.3	0.1	0:15.85	top
6756	0.0	20	0	715132	99164	74648	S	1.0	1.2	0:41.26	chrome
0	0.0	20	0	0	0	0	I	0.3	0.0	0:17.74	rcu sched
1174	0.0	20	0	188840	5320	4556	S	0.3	0.1	0:00.20	dcconf-serv+
1367	0.0	20	0	725822	98836	38180	S	0.3	1.2	0:10.36	cinnamon-s+
7570	0.0	20	0	594152	38668	27564	S	0.3	0.5	0:06.00	gnome-term+
8862	0.0	20	0	0	0	0	I	0.3	0.0	0:01.82	worker/ut+
9749	0.0	20	0	0	0	0	I	0.3	0.0	0:00.00	worker/1:1



... aux systèmes  
sur puce

# Programme

- 1) Programmation
- 2) Structures de données
- 3) Réseaux informatiques
- 4) Architecture matérielle & systèmes d'exploitation
- 5) Bases de données
- 6) Algorithmique

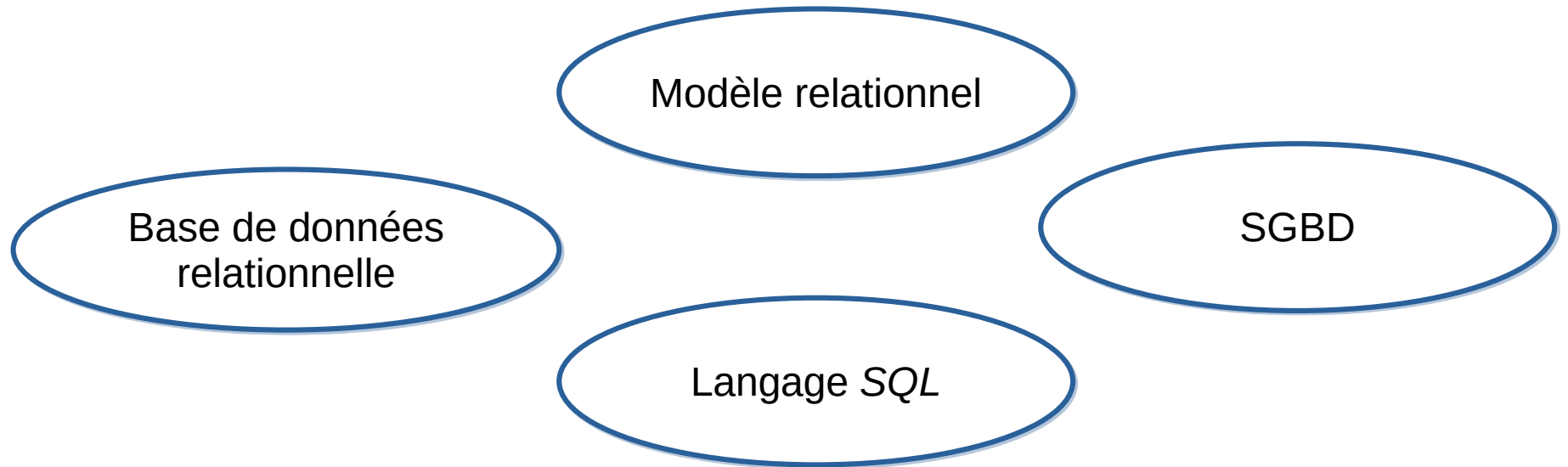


```
top - 11:32:00 up 2:46, 1 user, load average: 0.03, 1.08, 1.29
taskset: 230 total, 1 on cpu, 185 on wall, 0 mcpu, 0 mpage
VCPU(s): 10, 4 ut, 5.2 st, 6.8 ni, 82.9 id, 2.1 wa, 0.0 hi, 0.2 si, 0.0 st
KIB Mem: 3874128 total, 583124 free, 2392320 util, 328848 inactive
KIB ch: 2097140 total, 2090260 free, 740 util, 679586 dirty Mem
```

PID	UTL	PP	NI	VSZ	RES	SHR	S	CPUS	MEM	VSZ%	RES%	CPUS%	MEM%
1381	0.00	0	0	341180	14492	7000	S	21.0	1.0	5.17	90	0.00	0.00
889	0.00	0	0	50250	94320	67724	S	11.0	1.2	4.32	80	0.00	0.00
4245	0.00	0	0	2451700	112064	78012	S	5.3	1.4	2.38	84	0.00	0.00
1380	0.00	0	0	150400	17564	24884	S	4.6	0.4	0.86	48	0.00	0.00
6624	0.00	0	0	983932	163340	137072	S	4.3	3.3	6.17	45	0.00	0.00
8105	0.00	0	0	843420	32144	28840	S	4.0	0.4	2.42	12	0.00	0.00
7624	0.00	0	0	729540	181150	71624	S	3.8	1.3	0.82	89	0.00	0.00
1120	0.00	0	0	222080	37712	12064	S	2.8	0.2	3.12	20	0.00	0.00
5127	0.00	0	0	181580	29404	13704	S	1.7	3.4	3.44	68	0.00	0.00
4882	0.00	0	0	45580	4420	3528	R	1.3	0.1	6.15	45	0.00	0.00
8756	0.00	0	0	155100	99164	74044	S	1.0	1.2	0.61	35	0.00	0.00
0	0.00	0	0	0	0	0	T	0.3	0.0	0.17	74	0.00	0.00
1174	0.00	0	0	108940	5120	4556	S	0.3	0.1	0.86	20	0.00	0.00
1287	0.00	0	0	715822	93820	38188	S	0.3	1.2	0.18	30	0.00	0.00
2716	0.00	0	0	344120	1860	2704	S	0.3	0.1	0.86	48	0.00	0.00
8882	0.00	0	0	0	0	0	T	0.3	0.0	0.41	83	0.00	0.00
1340	0.00	0	0	0	0	0	T	0.3	0.0	0.86	88	0.00	0.00

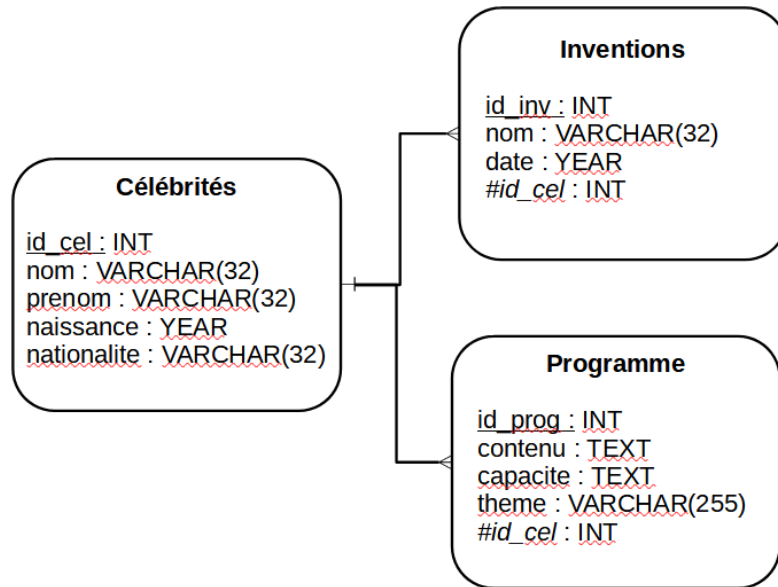


## 5- Bases de données

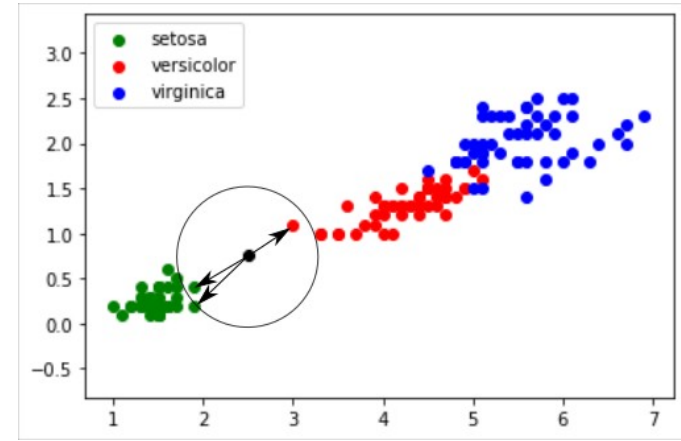


# Projet bases de données & IA

Conception et implémentation  
d'une **base de données**



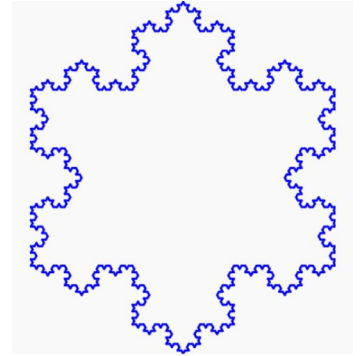
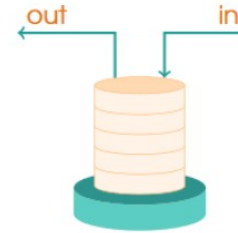
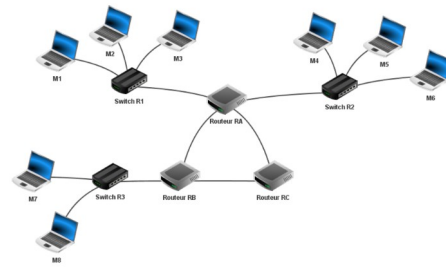
Implémentation d'un **algorithme d'IA** (k plus proches voisins, arbre de décision) pour faire des **prédictions**





# Programme

- 1) Programmation
- 2) Structures de données
- 3) Réseaux informatiques
- 4) Architecture matérielle & systèmes d'exploitation
- 5) Bases de données
- 6) Algorithmique



A screenshot of a terminal window displaying system information and a table of process data. The terminal output includes system statistics like CPU usage, memory usage, and disk I/O. Below this, a table lists various processes with columns for PID, PPID, USER, CPU, MEM, and COMMAND.

PID	PPID	USER	CPU	MEM	COMMAND
1	0	root	0.0	0.0	init
2	0	root	0.0	0.0	smd
3	0	root	0.0	0.0	smd
4	0	root	0.0	0.0	smd
5	0	root	0.0	0.0	smd
6	0	root	0.0	0.0	smd
7	0	root	0.0	0.0	smd
8	0	root	0.0	0.0	smd
9	0	root	0.0	0.0	smd
10	0	root	0.0	0.0	smd
11	0	root	0.0	0.0	smd
12	0	root	0.0	0.0	smd
13	0	root	0.0	0.0	smd
14	0	root	0.0	0.0	smd
15	0	root	0.0	0.0	smd
16	0	root	0.0	0.0	smd
17	0	root	0.0	0.0	smd
18	0	root	0.0	0.0	smd
19	0	root	0.0	0.0	smd
20	0	root	0.0	0.0	smd
21	0	root	0.0	0.0	smd
22	0	root	0.0	0.0	smd
23	0	root	0.0	0.0	smd
24	0	root	0.0	0.0	smd
25	0	root	0.0	0.0	smd
26	0	root	0.0	0.0	smd
27	0	root	0.0	0.0	smd
28	0	root	0.0	0.0	smd
29	0	root	0.0	0.0	smd
30	0	root	0.0	0.0	smd
31	0	root	0.0	0.0	smd
32	0	root	0.0	0.0	smd
33	0	root	0.0	0.0	smd
34	0	root	0.0	0.0	smd
35	0	root	0.0	0.0	smd
36	0	root	0.0	0.0	smd
37	0	root	0.0	0.0	smd
38	0	root	0.0	0.0	smd
39	0	root	0.0	0.0	smd
40	0	root	0.0	0.0	smd
41	0	root	0.0	0.0	smd
42	0	root	0.0	0.0	smd
43	0	root	0.0	0.0	smd
44	0	root	0.0	0.0	smd
45	0	root	0.0	0.0	smd
46	0	root	0.0	0.0	smd
47	0	root	0.0	0.0	smd
48	0	root	0.0	0.0	smd
49	0	root	0.0	0.0	smd
50	0	root	0.0	0.0	smd
51	0	root	0.0	0.0	smd
52	0	root	0.0	0.0	smd
53	0	root	0.0	0.0	smd
54	0	root	0.0	0.0	smd
55	0	root	0.0	0.0	smd
56	0	root	0.0	0.0	smd
57	0	root	0.0	0.0	smd
58	0	root	0.0	0.0	smd
59	0	root	0.0	0.0	smd
60	0	root	0.0	0.0	smd
61	0	root	0.0	0.0	smd
62	0	root	0.0	0.0	smd
63	0	root	0.0	0.0	smd
64	0	root	0.0	0.0	smd
65	0	root	0.0	0.0	smd
66	0	root	0.0	0.0	smd
67	0	root	0.0	0.0	smd
68	0	root	0.0	0.0	smd
69	0	root	0.0	0.0	smd
70	0	root	0.0	0.0	smd
71	0	root	0.0	0.0	smd
72	0	root	0.0	0.0	smd
73	0	root	0.0	0.0	smd
74	0	root	0.0	0.0	smd
75	0	root	0.0	0.0	smd
76	0	root	0.0	0.0	smd
77	0	root	0.0	0.0	smd
78	0	root	0.0	0.0	smd
79	0	root	0.0	0.0	smd
80	0	root	0.0	0.0	smd
81	0	root	0.0	0.0	smd
82	0	root	0.0	0.0	smd
83	0	root	0.0	0.0	smd
84	0	root	0.0	0.0	smd
85	0	root	0.0	0.0	smd
86	0	root	0.0	0.0	smd
87	0	root	0.0	0.0	smd
88	0	root	0.0	0.0	smd
89	0	root	0.0	0.0	smd
90	0	root	0.0	0.0	smd
91	0	root	0.0	0.0	smd
92	0	root	0.0	0.0	smd
93	0	root	0.0	0.0	smd
94	0	root	0.0	0.0	smd
95	0	root	0.0	0.0	smd
96	0	root	0.0	0.0	smd
97	0	root	0.0	0.0	smd
98	0	root	0.0	0.0	smd
99	0	root	0.0	0.0	smd
100	0	root	0.0	0.0	smd



## 6- Algorithmique

Comment trouver l'algorithme *le plus efficace* pour résoudre un problème ?

Programmation dynamique

Diviser pour régner

Recherche textuelle

Algorithmique sur les arbres,  
les graphes

# 6- Algorithmique

Comment trouver l'algorithme *le plus efficace* pour résoudre un problème ?

**Programmation dynamique**

Diviser pour régner

Recherche textuelle

Algorithmique sur les arbres,  
les graphes



## 6- Algorithmique

Comment trouver l'algorithme *le plus efficace* pour résoudre un problème ?

Programmation dynamique

Recherche textuelle

Algorithmique sur les arbres,  
les graphes

**Diviser pour régner**



## 6- Algorithmique

Comment trouver l'algorithme *le plus efficace* pour résoudre un problème ?

Programmation dynamique

Diviser pour régner

**Recherche textuelle**

Algorithmique sur les arbres,  
les graphes

## 6- Algorithmique

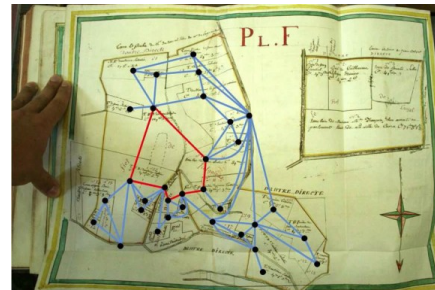
Comment trouver l'algorithme *le plus efficace* pour résoudre un problème ?

Programmation dynamique

Diviser pour régner

Recherche textuelle

**Algorithmique sur les arbres,  
les graphes**



# Perspectives

- Profil **NSI + Maths/Physique/SVT** :
  - Licence scientifique
  - BUT (informatique, R et T, STID,...)
  - Écoles d'ingénieur post-bac
  - Classe préparatoire MP2I (Maths Tle + Physique 1ère)
- Profil **NSI + SES** :
  - Licence (sciences éco,...)
  - BUT (information communication, MMI, QLIO...)
  - École de commerce
- D'autres combinaisons sont possibles



Questions ?