

# **Algorithmique et programmation**

**NF01**

**Philippe TRIGANO**

# Documentation

## ➤ Site web

- <http://www4.utc.fr/~nf01>

## ➤ Plate-forme pédagogique

- Moodle
  - ✓ <http://tice.utc.fr/moodle/course/view.php?name=NF01>
  - ✓ Accès possible à partir du site web de l'UV

## ➤ Polycopié

- Téléchargeable sur le site web
- Version papier reliée en vente à le BUTC

# Objectifs

## ➤ **Algorithmique**

- Déterminer une séquence d'actions primitives permettant de réaliser un traitement

## ➤ **Programmation**

- Coder à l'aide d'un langage, en respectant la syntaxe de ce langage

## ➤ **Langage**

- Pascal

# Langages

## ➤ Langage machine

- Les instructions sont des codes binaires
- Ex : 10110000 01100001

## ➤ Langage assembleur

- Les instructions sont de type STO, ADD, JMP, MOV
- Ex : mov %al,\$0x61

## ➤ Langages de programmation

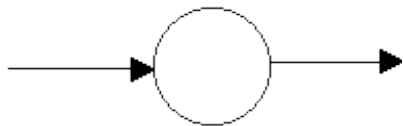
- Plus ou moins indépendants de la machine
- Ex : C, C++, Pascal, Java, Lisp, Prolog, ...
- Ex d'instructions :
  - ✓ if ... then ... else
  - ✓  $x := x + 1$

# Diagrammes de Conway



Catégorie syntaxique

*Permet de décrire la structure (ex: phrase, groupe nominal...)*

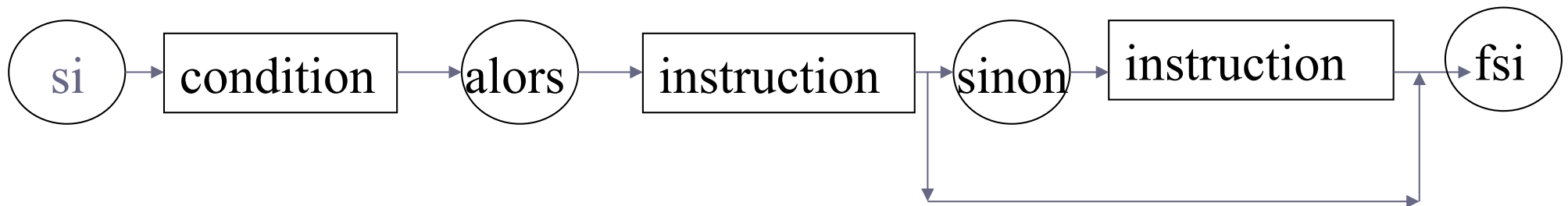


Symbole terminal

*Mot du langage (ex: chat, le, blanc, aime, Julie, dort, mange...)*

# Exemple

Instruction alternative



# Codage de l'information

## ➤ Dépend de la nature des données

- Vrai/Faux
  - ✓ 1 bit suffirait
- Caractères
  - ✓ Exemple : code ASCII
- Nombres
  - ✓ Divers codages possibles
- Images, sons
  - ✓ Echantillonnage, Numérisation

**Table ASCII  
(7 bits : 0-127)**

Ctl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
^@	0	00		NUL	32	20	sp	64	40	@	96	60	`
^A	1	01	☐	SOH	33	21	!	65	41	A	97	61	a
^B	2	02	⊠	SIX	34	22	"	66	42	B	98	62	b
^C	3	03	♥	EIX	35	23	#	67	43	C	99	63	c
^D	4	04	♦	EOI	36	24	\$	68	44	D	100	64	d
^E	5	05	♣	ENQ	37	25	%	69	45	E	101	65	e
^F	6	06	♠	ACK	38	26	&	70	46	F	102	66	f
^G	7	07	•	BEL	39	27	'	71	47	G	103	67	g
^H	8	08	◼	BS	40	28	(	72	48	H	104	68	h
^I	9	09	○	HI	41	29	)	73	49	I	105	69	i
^J	10	0A	◼	LF	42	2A	*	74	4A	J	106	6A	j
^K	11	0B	♂	VI	43	2B	+	75	4B	K	107	6B	k
^L	12	0C	♀	FF	44	2C	,	76	4C	L	108	6C	l
^M	13	0D	♪	CR	45	2D	-	77	4D	M	109	6D	m
^N	14	0E	♫	SD	46	2E	.	78	4E	N	110	6E	n
^O	15	0F	✱	SI	47	2F	/	79	4F	O	111	6F	o
^P	16	10	▶	SLE	48	30	0	80	50	P	112	70	p
^Q	17	11	◀	CS1	49	31	1	81	51	Q	113	71	q
^R	18	12	↕	DC2	50	32	2	82	52	R	114	72	r
^S	19	13	!!	DC3	51	33	3	83	53	S	115	73	s
^T	20	14	¶	DC4	52	34	4	84	54	T	116	74	t
^U	21	15	⚡	NAK	53	35	5	85	55	U	117	75	u
^V	22	16	▬	SYN	54	36	6	86	56	V	118	76	v
^W	23	17	⚙	EIB	55	37	7	87	57	W	119	77	w
^X	24	18	↑	CAN	56	38	8	88	58	X	120	78	x
^Y	25	19	↓	EM	57	39	9	89	59	Y	121	79	y
^Z	26	1A	→	SIB	58	3A	:	90	5A	Z	122	7A	z
^[	27	1B	←	ESC	59	3B	;	91	5B	[	123	7B	{
^\	28	1C	└	FS	60	3C	<	92	5C	\	124	7C	
^]	29	1D	⛶	GS	61	3D	=	93	5D	]	125	7D	}
^^	30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~
^_	31	1F	▼	US	63	3F	?	95	5F	_	127	7F	Δ†



# Exemple d'algorithme

## ➤ Objectif

- Déterminer une séquence d'actions primitives permettant de réaliser un traitement

## ➤ **Exemple** : calcul du périmètre d'un cercle de rayon $r$

## ➤ **Solution** :

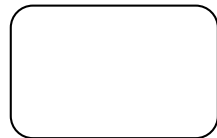
- Lire  $r$
- Calculer  $P := 2 * 3.14159 * r$
- Afficher  $P$

# Exemple

***Résolution de l'équation du premier degré :  $ax + b = 0$  dans  $R$  :***

- **1. lire les coefficients  $a$  et  $b$**
- **2. si  $a = 0$  alors**
  - si  $b = 0$  alors
  - afficher ("l'ensemble des solutions est  $R$ ")
  - sinon
  - afficher ( "pas de solution")
  - fin si
- sinon**
- solution  $\leftarrow -b / a$
- afficher ("La solution est : ", solution)
- fin si**

# Organigramme



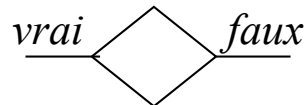
Début/fin



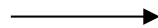
Entrées/Sorties



Traitement (séquence d'instructions)



Test



Sens de lecture

# Exemple

## *Calcul du salaire net*

### Données

SH : salaire horaire

NH : nombre d'heures

PR : % de retenues non  
plafonnées

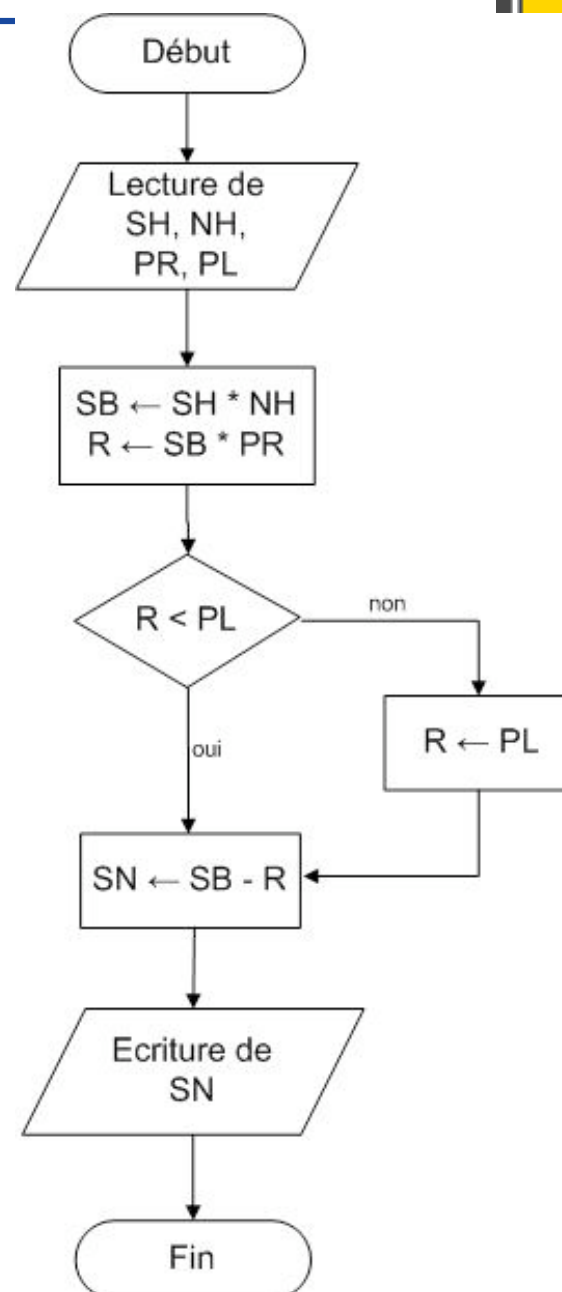
PL : plafond

### Variables

SB : salaire brut

SN : salaire net

R : retenues ( $R = SB * PR$ )



# Exercice

- **Ecrire un algorithme permettant de calculer la moyenne de  $n$  nombres.**
  - $n$  sera entré par l'utilisateur
  - les  $n$  nombres également

# Une première proposition

- Lire le nombre  $n$  de données
- Lire les  $n$  données
- Faire la somme  $s$  des données
- $\text{moyenne} \leftarrow s / n$
- Afficher moyenne

# Algorithme plus détaillé

- **soient**
  - $n$  et  $i$  des entiers,
  - *donnee*, *moyenne* et *somme* des réels
- **afficher (« Combien de données ? »)**
- **lire( $n$ )**
- **si  $n > 0$  alors**
  - $\text{somme} \leftarrow 0$
  - pour  $i$  allant de 1 à  $n$  faire
    - ✓ lire(*donnee*)
    - ✓  $\text{somme} \leftarrow \text{somme} + \text{donnee}$
  - $\text{moyenne} \leftarrow \text{somme} / n$
  - afficher ('la moyenne est ', *moyenne*)
- **sinon**
  - afficher('pas de de données')
- **fsi**

```

program moyenne ;           {En-tête}
var
  {Déclarations}
  donnee, somme, moyenne : real;
  i, n : integer ;

begin
  {début du bloc d'instructions}
  writeln('entrer le nombre de
    données');
  readln(n);
  if n > 0 then
    begin
      somme := 0;
      for i := 1 to n do
        begin
          read(donnee);
          somme := somme +
            donnee;
        end;
      moyenne := somme / n;
      writeln('moyenne =',moyenne);
    end
  else
    writeln('pas de donnees');
  end.

```

