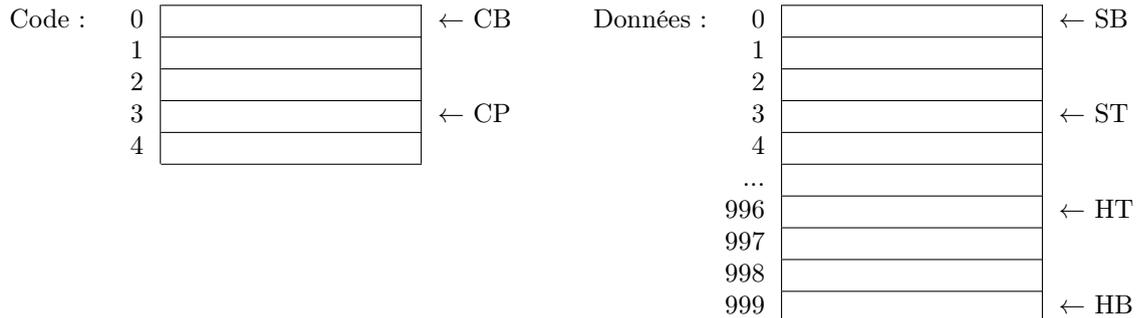


# La machine TAM

Machine à pile. Pas de registre de donnée.



Instructions :

PUSH n	ST = ST + n
POP (d) n	a = ST - d; ST = ST - d - n; Pour i de d à 0 Donnees(ST++) = Donnees[a++] fin pour
LOADL n	Donnees(ST) = n; ST = ST+1
LOADA d[r]	ST = d[r] ; ST = ST + 1;
LOAD (n) d[r]	Pour i de 0 a n-1 Donnees(ST+i) = Donnees(val(r)+d+i) fin pour; ST = ST+n
LOADI (n)	Empile n mots lus à l'adresse précédemment empilée
STORE (n) d[r]	Pour i de 0 a n-1 Donnees(val(r)+d+i) = Donnees(ST+i-n); fin pour; ST = ST-n
STOREI (n)	Ecrit les n mots empilés, à l'adresse empilée
JUMP etiq	CP = val(etiq)
JUMP d[r]	CP = val(r) + d
JUMPI	ST = ST - 1; CP = data[ST]
JUMPIF (n) etiq	si Donnees(ST - 1) = n alors CP = val(etiq) fin si; ST = ST - 1
JUMPIF (n) d[r]	si Donnees(ST - 1) = n alors CP = val(r) + d fin si; ST = ST - 1
SUBR op	Appel de op (fonction pré-définie - cf verso) consommation des arguments laissés en sommet de pile
CALL (r) op	Appel de op (fonction utilisateur) Donnees(ST) = r        --- Donnees(ST+1) = LB       Enregistrement d'activation Donnees(ST+2) = CP+1 --- LB = ST ST = ST + 3 CP = op
CALLI	Appel de la fonction dont l'adresse est en sommet de pile
RETURN (r) p	CP = Donnees(LB + 2) LB = Donnees(LB + 1) Pour i de 0 a p-1 Donnees(LB-p+r-1) = Donnees(ST-i); fin pour; ST = LB-p+r -- dépile p paramètres et garde r résultats
HALT	Arret

## Primitives prédéfinies de la machine TAM

Nom	Paramètres	Résultat	
Fonctions Booléens			
BNeg	1	1	Négation logique
BOr	2	1	Ou logique
BAnd	2	1	Et logique
BOut	1	0	Affiche sur <code>stdout</code> un booléen ( <code>true</code> ou <code>false</code> )
BIn	0	1	Lit sur <code>stdin</code> un booléen (1 ou 0)
B2C	1	1	Conversion vers un caractère ( <code>true</code> = '1', <code>false</code> = '0')
B2I	1	1	Conversion vers un entier ( <code>true</code> = 1, <code>false</code> = 0)
B2S	1	1	Conversion vers une chaîne de caractères (empile l'adresse de la chaîne dans le tableau des chaînes)
Fonctions Caractères			
COut	1	0	Affiche sur <code>stdout</code> un caractère
CIn	0	1	Lit sur <code>stdin</code> un caractère
C2B	1	1	Conversion vers un booléen ('0' = <code>false</code> , sinon <code>true</code> ).
C2I	1	1	Conversion vers un entier (le code ASCII)
C2S	1	1	Conversion vers une chaîne de caractères (empile l'adresse de la chaîne dans le tableau des chaînes)
Fonctions Entiers			
INeg	1	1	Négation entière
IAdd	2	1	Addition entière
ISub	2	1	Soustraction entière
IMul	2	1	Multiplication entière
IDiv	2	1	Diviseur dans division entière
IMod	2	1	Reste dans division entière
IEq	2	1	Test égalité entre 2 entiers
INeq	2	1	Test différence entre 2 entiers
ILss	2	1	Test inférieur strictement entre 2 entiers
ILEq	2	1	Test inférieur ou égal entre 2 entiers
IGtr	2	1	Test supérieur strictement entre 2 entiers
IGeq	2	1	Test supérieur ou égal entre 2 entiers
IOut	1	0	Affiche sur <code>stdout</code> un entier
IIn	0	1	Lit sur <code>stdin</code> un entier
I2B	1	1	Conversion vers un booléen (0 = <code>false</code> , sinon <code>true</code> , )
I2C	1	1	Conversion vers un caractère (le code ASCII). Non implémentée.
I2S	1	1	Conversion vers une chaîne de caractères (empile l'adresse de la chaîne dans le tableau des chaînes)

Nom	Paramètres	Résultat	
Fonctions Mémoires			
MVoid	0	1	Renvoie la valeur « adresse non initialisée »
MAlloc	1	1	Alloue un bloc mémoire et renvoie son adresse
MFree	1	0	Libère un bloc mémoire
MCompare	2	1	Test égalité entre le contenu de 2 blocs mémoire dont les adresses sont en sommet de pile
MCopy	3	0	Copie le contenu d'un bloc mémoire dans le second bloc mémoire. Dans l'ordre d'empilage : taille, adr destination, adr source
Fonctions Chaînes de caractères			
SAlloc	1	1	Crée une chaîne vide et empile son adresse dans le tableau des chaînes.
SFree	1	0	Libère l'espace de la chaîne
SCopy	1	1	Duplique la chaîne et empile l'adresse de la chaîne dupliquée
SConcat	2	1	Concatène deux chaînes (résultat dans le premier argument - gardé en sommet de pile)
SOut	1	0	Affiche sur <code>stdout</code> une chaîne
SIn	0	1	Lit sur <code>stdin</code> une chaîne
S2B	1	1	Conversion vers un booléen ("false", "f" ou "0" = <code>false</code> , sinon <code>true</code> , )
S2C	1	1	Conversion vers un caractère (le code ASCII du première caractère de la chaîne)
S2I	1	1	Conversion vers un entier (ne fait rien si la chaîne n'est pas un entier)