MVàP Cheat Sheet

Résumé des instructions

Code	Pile	sp	pc	Condition
PUSHI n	P[sp] := n	sp+1	pc+2	n est une valeur entière
ADD (SUB, MUL, DIV)	P[sp−2]:=P[sp−2] + P[sp−1]	sp-1	pc+1	2 entiers en sommet de pile
INF (INFEQ, SUP, SUPEQ, EQUAL, NEQ)	P[sp−2]:= 1 si P[sp−2] < P[sp−1], 0 sinon	sp-1	pc+1	2 entiers en sommet de pile
PUSHG n (PUSHL n)	P[sp] := P[gp+n] (P[sp] := P[fp+n])	sp+1	pc+2	n entier t.q. gp+n < sp
STOREG n (STOREL n)	P[gp+n] := P[sp-1] (P[fp+n] := P[sp-1])	sp-1	pc+2	n entier t.q. gp+n < sp
JUMP label		sp	instr(label)	
JUMPF label		sp−1	pc+2 si P[sp−1]≠0, instr(label) sinon	
				label correspond

CALL label		...	instr(label)	à une adresse dans le code
RETURN		...	...	
POP		sp−1	pc+1	sp > 1
HALT				
READ	P[sp] := entier lu	sp+1	pc+1	un entier sur l'entrée standard
WRITE		sp	pc+1	

Instructions supplémentaires

Code	Pile	sp	pc	Condition
PADD	P[sp−2] := P[sp−2] + P[sp−1]	sp−1	pc+1	adresse et entier en sommet pile
PUSHR n	P[sp−1] := P[P[sp−1] + n]	sp	pc+2	n entier, adresse en sommet pile
STORER n	P[P[sp−2] + n] := P[sp−1]	sp−2	pc+2	n entier, adresse en 2^e position

				sur la pile
FREE n		sp−n	pc+2	sp > n
ALLOC n	P[x] := 0 pour sp < x < sp+n	sp+n	pc+2	
JUMPI label		sp−1	instr(label) + P[sp−1]	entier en sommet de pile
DUP	P[sp]:=P[sp−1]	sp+1	pc+1	
PUSHF f	P[sp],P[sp+1] := f	sp+2	pc+3	f est une valeur en flottant
FADD (FSUB, FMUL, FDIV)	P[sp−2],P[sp−1]:= (P[sp−4],P[sp−3]) + (P[sp−2],P[sp−1])	sp−2	pc+1	2 flottants en sommet de pile
FINF (FINFEQ, FSUP, FSUPEQ, FEQUAL, FNEQ)	P[sp−4]:= 1 si (P[sp−4],P[sp−3]) < (P[sp−2],P[sp−1]), 0 sinon	sp−3	pc+1	2 flottants en sommet de pile
READF	P[sp],P[sp+1] := f	sp+2	pc+1	un flottant sur l'entrée standard
WRITEF		sp	pc+1	