

A	$\boxed{reg} \leftarrow \boxed{num} + \boxed{num}$	Sečte dvě čísla.
C	CALL \boxed{dir}	Volání podprogramu. Uloží aktuální souřadnice a směr na zásobník, načtež zabočí zadaným směrem. Pokud už je na zásobníku 1 000 položek, spadne.
D	$\boxed{reg} \leftarrow \boxed{num} / \boxed{num}$	Vydělí dvě čísla. Při dělení nulou spadne.
E	IF $\boxed{num} = \boxed{num}$	Provede následující instrukci, pokud je podmínka splněna. Jinak ji přeskočí.
F	FACE \boxed{azi}	Otočí se zadaným absolutním směrem (na sever/jih/východ/západ).
G	GOTO \boxed{lab}	Skočí na zadané návěští. Pokud ho nenajde, skončí běhovou chybou. Zachová směr provádění programu.
H	HALT	Zastaví vykonávání programu.
I	INIT	Chová se stejně jako FACE SOUTH a navíc určuje, že na tomto místě program začíná. Tato instrukce se v programu musí vyskytovat právě jednou.
J	JUMP	Přeskočí následující instrukci.
K	RETURN	Návrat z podprogramu. Vybere ze zásobníku pozici a směr uložené instrukcí CALL . Pokud je zásobník prázdný, spadne.
L	LABEL \boxed{lab}	Poloha této instrukce definuje návěští. Každé návěští se v programu smí nacházet nejvýše jednou.
M	$\boxed{reg} \leftarrow \boxed{num} * \boxed{num}$	Vynásobí dvě čísla.
N	$\boxed{reg} \leftarrow \mathbf{ARR} \boxed{num}$	Vytvoří pole zadané velikosti vyplněné nulami. Velikost je nezáporné číslo, maximálně 1 000 000. Odkaz na pole uloží do registru.
P	$\boxed{reg} \leftarrow 10 \boxed{num} + \boxed{num}$	Přidá k číslu další číslici (přesněji řečeno, první číslo vynásobí 10, přičte k němu druhé a výsledek uloží do registru).
QL	$\boxed{reg} \leftarrow \boxed{num} \ll \boxed{num}$	Desítkový posun doleva. Druhý argument vynásobí desítkou tolikrát, kolik je třetí argument (číslo mezi 0 a 10).
QR	$\boxed{reg} \leftarrow \boxed{num} \gg \boxed{num}$	Desítkový posun doprava. Druhý argument vydělí desítkou tolikrát, kolik je třetí argument (číslo mezi 0 a 10).
R	$\boxed{reg} \leftarrow \boxed{arr} [\boxed{num}]$	Čtení z pole: do registru uloží číslo, které v daném poli leží na dané pozici. Pokud indexujeme mimo rozsah, program spadne.
S	$\boxed{reg} \leftarrow \boxed{num} - \boxed{num}$	Odečte dvě čísla.
T	TURN \boxed{dir}	Otočí směr vykonávání programu.
U	IF $\boxed{num} \neq \boxed{num}$	Provede následující instrukci, pokud je podmínka splněna. Jinak ji přeskočí.
V	$\boxed{reg} \leftarrow \boxed{any}$	Uloží do registru číslo nebo ukazatel na pole.
W	$\boxed{arr} [\boxed{num}] \leftarrow \boxed{num}$	Zápis do pole. Pokud indexujeme mimo rozsah, program spadne.
X	IF $\boxed{num} < \boxed{num}$	Provede následující instrukci, pokud je podmínka splněna. Jinak ji přeskočí.
Y	IF $\boxed{num} > \boxed{num}$	Provede následující instrukci, pokud je podmínka splněna. Jinak ji přeskočí.
Z	$\boxed{reg} \leftarrow \boxed{num} \% \boxed{num}$	Vypočítá zbytek po dělení. Pokud dělíte nulou, program spadne. Pokud dělíte záporným číslem, výsledek není definován. Pokud dělíte záporné číslo kladným, vyjde nezáporný výsledek (jako v Pythonu).

\boxed{reg}	registr A–Z
\boxed{num}	registr nebo číslo 0–9
\boxed{arr}	registr A–Z ukazující na pole
\boxed{any}	číslo nebo registr obsahující číslo nebo ukazatel
\boxed{azi}	azimut NORTH/SOUTH/WEST/EAST
\boxed{dir}	směr LEFT/RIGHT/BACK
\boxed{lab}	návěští A–Z

Procesor počítá s 32-bitovými čísly, přetékaají jako v Céčku.
Pole jsou indexována od nuly, odkazujeme se na ně ukazateli.
Registry se jmenují A až Z, obsahují číslo nebo ukazatel.
Instrukce nejdříve čte vstupy, teprve pak zapisuje výstupy.
Maximální době běhu: 10^8 instrukcí.
Maximální velikost pole: 10^6 buněk.
Program je torus 20×10 .
Registry i pole jsou inicializována vstupem nebo nulami.