

inp w1 mul x 0 add x z mod x 26 div z 1 add x 14 eq1 x w x=0 (w<=9) eq1 x 0 x=1 mul y 0 y=0 add y 25 y=25 mul y x y=25 add y 1 y=26 mul z y z*=26 mul y 0 y=0 add y w y=w add y 12 y=w+12 mul y x y=w+12 add z y z+=(w+12) (push w1 + 12)	inp w4 mul x 0 add x z mod x 26 x=z%26 (ie x=top of stack) div z 26 (pop) add x -8 x -= 8 eq1 x w4(want x=w4) pop-8=w4 eq1 x 0 (want x!=0) mul y 0 add y 25 mul y x add y 1 mul z y add y 0 mul y w add y 3 mul y x (want x=0) add z y	inp w7 mul x 0 add x z mod x 26 div z 1 add x 14 eq1 x w eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y z*=26 mul y 0 add y w add y 10 mul y x add z y z+=(w+10) (push w7 + 10)	inp w10 mul x 0 add x z mod x 26 x=z%26 div z 26 (pop) add x -1 eq1 x w (want x=w10) pop-1=w10 eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y mul y 0 add y w add y 10 mul y x (want x=0) add z y	inp w13 mul x 0 add x z mod x 26 x=z%26 div z 26 (pop) add x -16 eq1 x w (want x=w13) pop-16=w13 eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y mul y 0 add y w add y 6 mul y x (want x=0) add z y
inp w2 mul x 0 add x z mod x 26 div z 1 add x 10 eq1 x w eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y z*=26 mul y 0 add y w add y 9 mul y x add z y z+=(w+9) (push w2 + 9)	inp w5 mul x 0 add x z mod x 26 div z 1 add x 11 eq1 x w eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y z*=26 mul y 0 add y w add y 0 mul y x add z y z+=(w+0) (push w5 + 0)	inp w8 mul x 0 add x z mod x 26 x=z%26 div z 26 (pop) add x -11 eq1 x w (want x=w8) pop-11=w8 eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y mul y 0 add y w add y 13 mul y x (want x=0) add z y	inp w11 mul x 0 add x z mod x 26 x=z%26 div z 26 (pop) add x -8 eq1 x w (want x=w11) pop-8=w11 eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y mul y 0 add y w add y 10 mul y x (want x=0) add z y	inp w14 mul x 0 add x z mod x 26 x=z%26 div z 26 (pop) add x -6 eq1 x w (want x=w14) pop-6=w14 eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y mul y 0 add y w add y 5 mul y x (want x=0) add z y
inp w3 mul x 0 add x z mod x 26 div z 1 add x 13 eq1 x w eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y z*=26 mul y 0 add y w add y 8 mul y x add z y z+=(w+8) (push w3 + 8)	inp w6 mul x 0 add x z mod x 26 div z 1 add x 11 eq1 x w eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y z*=26 mul y 0 add y w add y 11 mul y x add z y z+=(w+11) (push w6 + 11)	inp w9 mul x 0 add x z mod x 26 div z 1 add x 14 eq1 x w eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y z*=26 mul y 0 add y w add y 3 mul y x add z y z+=(w+3) (push w9 + 3)	inp w12 mul x 0 add x z mod x 26 x=z%26 div z 26 (pop) add x -5 eq1 x w (want x=w12) pop-5=w12 eq1 x 0 mul y 0 add y 25 mul y x add y 1 mul z y mul y 0 add y w add y 14 mul y x (want x=0) add z y	Notice there are 7 unconditional pushes and 7 unconditional pops. We must prevent the 7 conditional pushes that occur in the pop blocks by forcing x=0. This way the stack is empty (z=0) after all 14 blocks are run.

```
X (push w1 + 12)
X (push w2 + 9)
X (push w3 + 8)
X pop-8=w4
X (push w5 + 0)
X (push w6 + 11)
X (push w7 + 10)
X pop-11=w8
X (push w9 + 3)
X pop-1=w10
X pop-8=w11
X pop-5=w12
X pop-16=w13
X pop-6=w14
```

```
(w3 + 8) - 8 = w4
(w7 + 10) - 11 = w8
(w9 + 3) - 1 = w10
(w6 + 11) - 8 = w11
(w5 + 0) - 5 = w12
(w2 + 9) - 16 = w13
(w1 + 12) - 6 = w14
```

```
max sol: pick
highest values for
w1, w2,...
```

```
min sol: pick lowest
values for w1,
w2,...
```