

```

PC := 1;
Next:
case Code[PC] of
  save n: stack[n] := stack[top--]
  load n: stack[++top] := stack[n]
  negate: stack[top] := -stack[top]
  not: stack[top] := not stack[top]
  add: t := stack[top--]; stack[top] := stack[top] + t
  subtract: t := stack[top--]; stack[top] := stack[top] - t
  multiply: t := stack[top--]; stack[top] := stack[top] * t
  divide: t := stack[top--]; stack[top] := stack[top] / t
  mod: t := stack[top--]; stack[top] := stack[top] mod t
  or: t := stack[top--]; stack[top] := stack[top] or t
  and: t := stack[top--]; stack[top] := stack[top] and t
  equal: t := stack[top--]; stack[top] := stack[top] = t
  lessthan: t := stack[top--]; stack[top] := stack[top] < t
  greaterthan: t := stack[top--]; stack[top] := stack[top] > t
  read: stack[++top] := getinteger(input)
  print: putinteger(stack[top--])
  lit n: stack[++top] := n
  goto n: PC := n; goto Next
  iffalse n: if stack[top--] = 0 then PC:=n; goto Next fi
  iftrue n: if stack[top--] = 1 then PC:=n; goto Next fi
  stop: halt
end;
++PC;
goto Next;

```