1. let regex = {

canadianPostalCode: /^[ABCEGHJ-NPRSTVXY]\d[ABCEGHJ-NPRSTV-Z] \d[ABCEGHJ-NPRSTV-Z]\d$/,

visa: /^4\d{12}(\d{3})?$/,

masterCard: /^(5[1-5]\d{14}|222[1-9]\d{12}|22[3-9]\d{13}|2[3-6]\d{14}|27[01]\d{13}|2720\d{12})$/,

notThreeEndingInOO: /^(?![\p{L}]oo$)\p{L}*$/iu,

divisibleBy16: /^(0|00|000|[01]*0000)$/,

eightThroughThirtyTwo: /^([89]|[12]\d|3[0-2])$/,

notPythonPycharmPyc: /^(?!pyc$|python$|pycharm$)\p{L}*$/u,

restrictedFloats: /^[+-]?(\d+\.\d*|\d*)([eE][+-]?\d{1,3})$/,

palindromes2358:/^(?:([abc])\1|([abc])[abc]\2|([abc])([abc])[abc]\4\3|([abc])([abc])([abc])([abc])\8\7\6
\5)$/

pythonStringLiterals:
/^([fF]?)((?<!\\)'([^'\\\n]*(\\.[^'\\\n]*)*)'|(?<!\\)"([^"\\\n]*(\\.[^"\\\n]*)*)"|'''([^'\\]*(\\.|[^'\\])*?)'''|"""
([^"\\]*(\\.|[^"\\])*?)"""|'''abc\\''')$/

}

2. WebAssembly:

```
f:
     local.get       0   ;; load input
     i32.const       3   ;; push 3
     i32.mul             ;; 3 * input
     i32.const       1   ;; push 1
     i32.add             3 * input + 1
     local.get       0   ;; load input
     i32.const       1   ;; push 1
```

```
    i32.shr_s
    local.get       0   ;; load input
    i32.const       1   ;; push 1
    i32.and             ;; 1 if odd, 0 if even
    i32.select          ;; return either n / 2 or 3 * n + 1
    end_function
```

x86-64:

```
f:
push    rbp ; save base pointer
mov     rbp, rsp ; rsp to rbp
mov     DWORD PTR [rbp-4], edi
mov     eax, DWORD PTR [rbp-4]
and     eax, 1 ; check parity
test    eax, eax
jne     .L2 ; jump if eax != 0
mov     eax, DWORD PTR [rbp-4]
mov     edx, eax ; eax to edx
shr     edx, 31
add     eax, edx
sar     eax
jmp     .L4 ; jump to end
.L2:
mov     edx, DWORD PTR [rbp-4]
mov     eax, edx ; edx into eax
add     eax, eax ; eax * eax
add     eax, edx ; eax * eax * eax
```

```
add     eax, 1 ; add 1
.L4:
pop     rbp ; pop from stack
ret ; return eax
```

3. Assume it is decidable. Then there is a TM that decides whether L(M1)=L(M2), meaning it outputs either yes or no. Make M1 = to a machine that accepts M and w as input and M replaces its input with w if it accepts w and leaves it blank otherwise. Make M2 = to a reject machine that rejects anything. So, in some cases, L(M1)=L(M2) (when M1 doesn't accept w) and in other cases they do not equal. This means we decided the halting problem which we know is undecidable so there is a contradiction meaning the original language is undecidable.