```
bits 32
global start
extern exit
import exit msvcrt.dll
segment data use32 class=data
    s1 dw 55h, 32h, 114h
    s2 dw 0xFFFFFF35, 34h, 110h
    l equ ($-s2)/2
    s3 resb l

segment code use32 class=code
  start:
    mov ECX, l; ECX = lungimea sirurilor
    JECXZ saredacasirulezero
      mov ESi, 0; counter ESi = 0
      repeta:
        mov AX, [S1+ESi]; ~~in AX~~ pune in AX al ESi-lea
        element din s1 (primul fiind 0)
        mov BX,[S2+ESi] ; -||- BX -||- din s2
        cmp AX, BX ; compară AX cu BX
          JG axmaimarecabx ; daca AX este mai
          mare in interpret cu semn
            JL bxmaimarecaax
              JMP sarealtfel
          axmaimarecabx:
            mov [S3+ESi], AX; il adaugă pe AX
            in sirul s3 pe pozitia ESi
```

Scanned with CamScanner

```
bxmaimarecaax:
    mov [s3+ESI],BX ; il adaugă pe BX
    in sirul s3 pe pot. ESI

  sarealtfel:
     shl AX, 8 ; elem1 va deveni cuvântul
     superior al dublucuv. din sir1
     mov BH,0 ; elem2 va deveni cuvântul
     inferior al dublucuvântului din sir2
     mov DL, AL;
     mov AL,BL;   am interschimbat val.
     pt. împărțire elem 2 la elem1
      => DH = elem1 și AH = elem2
     mov AH,0 ; AL -> AX
     div DL ; in AL avem rezultatul
     împărțirii AX/DL
     mov AX,0 ; fac din nou conversia la
                       word
     mov [S3+ESI], AX ; il pun pe AX pe
                                        S3
     inc ESI
   loop repeta

Saredacăsuntezero:


push   dword 0
call   [exit]
```