

```
#Scrivere un programma assembler che si comporta come segue:
#
#1. legge da tastiera un numero naturale A in base 10,
#   sotto l'ipotesi che A stia su 32 bit
#2. se A e' nullo, termina, altrimenti:
#3. stampa A in tutte le basi da 2 a 9 su righe successive,
#   premettendo a ciascuna sequenza di cifre la base cui si riferisce
#4. torna al punto 1.
#
#Esempio:
#?00065536
#2: 1000000000000000
#3: 10022220021
#4: 10000000
#5: 4044121
#6: 1223224
#7: 362032
#8: 20000
#9: 108807
```

#-----

```
A:          .FILL 1,4

_main:      NOP

punto1:     CALL newline
            MOV $'?', %AL
            CALL outchar
            CALL indecimal_long
            CALL newline

            MOV %EAX, A
punto2:     CMP $0, %EAX
            JE fine_prog

# ESI contiene la base, CX il numero di cifre da stampare

punto3:     MOV $2, %ESI

ciclo_est:  CMP $9, %ESI
            JA punto1

            MOV %SI, %AX
            ADD $'0', %AL
            CALL outchar
            MOV $':', %AL
            CALL outchar
            MOV $' ', %AL
            CALL outchar
            MOV A, %EAX
            MOV $0, %EDX
            MOV $0, %CX

#mette nello stack le cifre di A in base SI, tramite divisioni successive

ciclo_int:  DIV %ESI
            PUSH %DX
            MOV $0, %EDX
            INC %CX
            CMP $0, %EAX
            JNE ciclo_int

ciclo_st:   POP %AX
            ADD $'0', %AL      # conversione da cifra a codifica ASCII
            CALL outchar
            DEC %CX
            JNZ ciclo_st

            INC %SI
            CALL newline
            JMP ciclo_est

fine_prog:  RET

.INCLUDE "C:/amb_GAS/utility"
```