

```

1 .GLOBAL _main
2 .INCLUDE "C:/amb_GAS/utility"
3
4 .DATA
5 vettore1: .FILL 6, 1, 0
6 vettore2: .FILL 1, 1, 0
7
8 .TEXT
9 _main:      MOV $6, %CL
10            LEA vettore1, %EDI
11            CALL richiedinnumero
12
13            # Richiedo Y
14            MOV $1, %CL
15            LEA vettore2, %EDI
16            CALL richiedinnumero
17
18            # Verifico che Y \neq 0, se e' uguale termino
19            CMP $0, vettore2
20            JE fine_male
21
22            CALL newline # Si chiede di lasciare una riga bianca
23
24 divisione:  MOV $0, %ESI # Dovro' scorrere il vettore
25            MOV vettore2, %CL # Divisore, uguale in ogni caso
26            MOV $6, %CH # Numero di divisioni da fare, variabile contatore
27            MOV $0, %AL
28
29 ciclo_div:  CALL moltiplicatore
30            # Ho in BX il numero moltiplicato per 10
31            MOV vettore1(%ESI), %AL
32            ADD %BL, %AL
33
34            MOV $0, %AH
35
36            PUSH %AX
37
38            CALL outdecimal_byte
39
40            MOV $'/', %AL
41            CALL outchar
42
43            MOV %CL, %AL
44            CALL outdecimal_byte
45
46            MOV $':', %AL
47            CALL outchar
48
49            MOV $' ', %AL
50            CALL outchar
51
52            POP %AX
53
54            # A questo punto ho in AX il dividendo
55            # Il divisore e' in CL
56            DIV %CL
57            # Dovrei avere quoziente in AL e resto in AH
58
59            PUSH %AX
60
61            MOV $'q', %AL
62            CALL outchar
63
64            MOV $'=', %AL
65            CALL outchar
66
67            POP %AX
68
69            CALL outdecimal_byte

```

```

70
71         PUSH %AX
72
73         MOV $',', %AL
74         CALL outchar
75
76         MOV $' ', %AL
77         CALL outchar
78
79         MOV $'r', %AL
80         CALL outchar
81
82         MOV $'=', %AL
83         CALL outchar
84
85         MOV %AH, %AL
86         CALL outdecimal_byte
87
88         POP %AX
89
90         INC %ESI
91
92         MOV %AH, %AL # Metto in AL il resto
93
94         DEC %CH
95         CMP $0, %CH
96
97         CALL newline
98         JNE ciclo_div
99
100 fine_bene:    CALL newline
101              JMP _main
102 fine_male:    XOR %EAX, %EAX
103              RET
104
105 # Sottoprogramma con cui viene richiesto un numero naturale in base 10
106 # Input: CL (numero di cifre richiesto), EDI (puntatore a memoria)
107 # Output: contenuto del vettore puntato da EDI aggiornato
108 richiedinnumero: CLD
109 ciclo:        CALL inchar # Valore in AL
110              CMP $'0', %AL
111              JB ciclo
112              CMP $'9', %AL
113              JA ciclo
114              CALL outchar
115              AND $0x0F, %AL # Converto da codifica ASCII a numero
116              STOSB # Aggiorno il vettore che contiene il numero
117 rn_poi:       DEC %CL
118              CMP $0, %CL
119              JNE richiedinnumero # Esco solo dopo aver compiuto il numero di
              cicli indicato in CL
120 fine_rn:      CALL newline
121              RET
122
123 # Sottoprogramma per la moltiplicazione
124 # Input: AL, cifra da moltiplicare
125 # Output: BX, cioe' AL moltiplicato per 10
126 moltiplicatore: PUSH %AX
127                PUSH %DX
128
129                MOV $10, %DL
130                MUL %DL
131
132                MOV %AX, %BX
133
134 fine_moltiplic: POP %DX
135                POP %AX
136                RET
137

```