**MUL-DIV**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Program A | Program B | Program C | Program D | Program E |  |  |
| Mov BL,250  Mov AL,244  Imul BL  Mov DH,AH  Mov AH,2  Mov DL,AL  Int 21h  Mov DL,DH  Int 21h  Mov AH,76  Int 21h | MovAX,5497  Mov BL,100  Div BL  Mov DL,AL  Mov DH,AH  Mov AH,2  Int 21h  Mov DL,DH  Int 21h  Stop | Mov AX,2000  Mov BX,150  Mul BX  Mov CX,AX  Mov AH,2  Int 21h  Mov DL,CH  Int 21h  Mov DL,CL  Int 21h  Stop | Mov DX,40  Mov AX,4000  Mov BX,10000  Div BX  Mov CX,AX  XCHG DH,DL  Mov AH,2  Int 21h  Print DH,CH,CL | Mov AH,1  Int 21h  Mov BL,2  Mul BL  Mov DL,AL  Mov AH,2  Int 21h  Mov AH,76  Int 21h  End |  |  |

Program A: BL=6 AL=12 BL\*AL AX=6\*12=72(H)

BL=250=-6 AL=244=-12 IMUL BL does AX=AL\*BL=(-6)(-12)=72=0\*256+72 AH=0 AL=72.

Replace IMUL by MUL AX=250\*244=61000 AH=238 AL=72 206\*6=1236=4,212 or -300=254,212

Program B: Div BL performs AL=AX div BL=54 AH=AX mod BL=97 Hence output is 6a.

BL: 156 o/p 35,37(#%) 5497 div 156=35 5497 mod 156=37

Idiv BL: BL=156=-100 5497=(-54)\*(-100)+97. Output 202,97 Since -54=202

AX=60139=-5397 BL:100 -5397=(-53)\*100-97 o/p 203,159 BL:156 -5397=(53)\*(-100)-97 o/p53,159

Program C: Mul BX is AX\*BX stored in DX:AX 2000\*150=3,00,000

DX=3,00,000 div 65536=4 AX=3,00,000 mod 65534=37856 Now AH=147 AL=224

Program D: Div BX is AX=DX:AX div BX DX=mod DX=40 AX=4000 DX:AX=65536\*40+4000 = 2625440 div-mod10000 AX=262 DX=5440 AH=1 AL=6 DH=21 DL=64

Program E: Read a letter output a letter with double ascii code. 2d #F

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| AL | BL | | Mul BL (Unsigned) | IMul BL Signed multiplication | | |
| 156 | 200 | | AX=31200 | AX=5600 because 156 is -100. 200 is -56. (-100)(-56)=+5600 | | |
| 156 | 100 | | AX=15600 | AX=55536 because 156 is -100. 100 is +100.  (-100)(+100)=-10000 it is (-10000+65536=55536) | | |
| AX | BL | Div BL (Unsigned) | | | IDiv BL Signed division |
| 12035 | 100 | AL=120 AH=35 | | | AL=120 AH=35 because 12035=(+100)(+120)+35 |
| 12035 | 156 | AL=77 AH=23  12035=77\*156+23 | | | AL=136 AH=35 because 12035=(-100)(-120)+35  0..32765 are positive and 32768..65535 are negative. |
| 53501 | 100 | divide overflow  535 unrepresentable | | | AL=136 AH=221 because -12035=(+100)(-120)-35  -120 is 256-120=136 -35 is 256-35=221 |
| 53501 | 156 | divide overflow | | | AL=120 AH=221 because -12035=(-100)(+120)-35 |

* 1. Read two digits. Find product. Input 98 output H. Input 76 output \* Input 99 output Q
  2. Read two letters. Find (product of ascii) divmod 100. Input FQ output 8F 70\*81=5670 EG0c
  3. Read a letter. Print last digit of ascii code. H2 b8. Second last digit. G7 g0 z2
  4. Read a letter Find (7\*ascii) div 5. 2F (50\*7=350) Gc
  5. (ascii-80)\*13+75 Sr N1 (ascii-80)\*(-13)+75 S$ Ne
  6. [1000 divmod (ascii-80)]+75 7#K pjS 2\*U (-1000 divmod..) 7sK n\*A
  7. [{122\*(ascii-80)} div 500]+75 (byte mul) iQ 7E (Hard)
  8. [{162\*(ascii-80)} div 500]+75 iS 7C
  9. [{1000 div (ascii-80)} mod 50] +75 YV G@ (unsign AX=AL is AH=0 sign AX=AL is CBW)
  10. [(ascii80) divmod 6]+75 (CBW:AX=AL) dNM 3GF [(ascii80) divmod -6]+75 dHM 3OF
  11. [100000 mod (x-80)]+75 nU 2U BW (use CBW) [-100000…] nA dK B?
  12. [100000 div {100\*(x-80)}]+75 2\* xd [-100000 div..] n\* x2
  13. [{(ascii-80)\*256} divmod 100]+75 LA3 E /; T Uc (no mul) divmod -100 LU3 Eg; TAc
  14. [{100\*(ascii-80)} divmod 256]+75 (no div) ZN3 FGc(not H) (using div) ZN3 FHc
  15. [100000 div {256\*(ascii-80)}]+75(no mul) d^ 7< sV A1(49) -100000 d8 7Z s@ Ae

[{65536\*(ascii-80)} div 10000] +75 (no mul) L1(49) Te