

Flag

Program A	Program B
.model small	Mov BL,100
.code	Add BL,120
Mov CL,100	Jmp g
Mov BH,105	Pushf
Mov AH,2	Pop AX
Mov DL,65	Or AL,00000001h
Add CL,BH	Push AX
JS L1	Popf
Add DL,3	g: Mov AH,2
L1: Int 21h	JC t
Mov AH,76	Mov DL,65
Int 21h	Jmp k
End	t: Mov DL,66
	k: Int 21h
	Stop

Program A: outputs 'A'. $100+105=205 > 127$. It is a negative number. Hence flag S=1.
 Let us use 205 in place of 105.
 $205+100=305 > 255$ hence $305-256=49=+49$ S=0
 Hence JS does not perform jump. output is 'D'.
 replace JS by JO.
 $100+105=205 \Rightarrow (-100)+(+105)=-51$ Wrong
 A wrong answer is called over flow. Hence flag 'Ov=1' Hence output 'A'.
 $140+105=245 \Rightarrow (-116)+(+105)=-11$. It is correct. Flag Ov=0 Hence 'D'.
 $150+160=310 > 254 \Rightarrow (-106)+(-96)=-202$ Wrong
 $200+220=420 > 254 \Rightarrow (-56)+(-36)=-92$ Correct
 $100-102=-2 > 254 \Rightarrow (+100)+(-102)=-2$ Correct
 $130-5-125=-125 \Rightarrow (-126)+(+5)=-121$ Wrong

Replace JS by JC. $100+105$ is no carry. $100+205$ is carry.

Replace JS by JP. See addition of 12&15 and 12&14

$12+15=27=11011b$ has 4(1's) even 1's Parity flag P=1 output A

$12+14=26=110103(\text{odd})(1's)$ P=0 o/pD $4+14=18=100102(\text{even})(1's)$ P=1 o/pA

Program B: Observe the output of the given program. Observe output by removing Jmp g.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Ov	D	I	T	S	Z		Ac		P		C

- Input 2 letters. Output 'A' if there is auxiliary carry in the addition of ascii codes. The output is 'B' otherwise. (89 → A) (82 → B) (gf → B) (km → A)
- Input 2 letters (ascii x and y). Output 'A' if (150+x) and (150+y) both have carry in their addition. Output 'B' otherwise. Use only one jump. [Hint: use and, pushf, popf] (af → B) (xy → A) (xa → B) (bz → B)
- Input a letter (ascii x). Output 'A' if (150+x) has carry. 'B' is outputted otherwise. Do not use JC. Use only JP. (z → A) (c → B).
- 'B' if (150+x) has carry. 'A' is outputted otherwise. No Jump. (z → B) (c → A). [Hint: Adc]
- 'A' if (150+x) has carry. 'B' is outputted otherwise. No jump. (z → A) (c → B)
- 'A' if (50+x) has overflow. 'B' is outputted otherwise. No jump. (z → A) (3 → B)
- 'A' when between 50 and 100. Output 'B' otherwise. use only mov, add, jo(once), int. Size < 15 lines (including .model small .code end).
- Input 2 letters (x and y). Output A if (x+70 < y+10). Output B otherwise. Do not use JG or JL. Use only JO and JS. [Definition: L=Ov ⊗ S, exactly one of overflow or sign flag] (2s → A) (2a → B) (2z → B) (d2 → A) (z2 → A) (dz → B) (<z → A) [<:60]
- Input a letter. Increment it. No shift, rotate, etc. add, sub. (adc 0) can be used.
- Input a letter. Increment it only when it is odd. (above restrictions) Hint: use and/or.
- Input letter (x). Output a letter whose binary representation is 010S00Ac0. Here S and Ac are sign and auxiliary flags respectively in (x+166). (A → P)(M → R)(a → @)(m → B).
- Do it for 0100S0Ac0. (A → H) (M → J) (a → @) (m → B).
- Read a letter. Output 'A' when ascii between 50 and 100. B otherwise. use only mov, int, cmp, pushf, popf, xor, jc
- In following programs: byte size, Mov, Int, All shift-rotate, PushF, PopF, STC, CLC
 - abcdefgh → abcdefh. (A → C)(B → @)(C → C)(D → D)(E → G)
 - abcdefgh → abcdefhg. (A → B)(B → A)(C → C)(D → D)(E → F)
 - abcdefgh → abcdefh0. (A → B)(B → @)(C → B)(D → D)(E → F)
 - abcdefgh → abc1100h (A → Y)(5 → 9)(3 → 9)(N → X)(; → 9)
 - abcdefgh → abefedg'h. (K → a)(L → r)(M → s)(N → p)(O → q)(5 → (ascii 31))

15. Write a program to print AAAAA. Use only Mov, add, int and one (once) among JL, JG, JC, JNC, JA, JB, JO, JNO, JP, JS, JNE