Trace the output of the following programs using the blanks to the right.

```
def f1(q):
    print("f1:", q)
    f2(q+2)
    print("f1:", q)
def f2(q):
    print("f2:", q)
def f3(q):
    print("f3:", q)
    f1(q+1)
    print("f3:", q)

f3(3)
```

```
    def f1(q):
        print ("1:", q)
        q += 1
        print ("1:", q)
    def f2(q):
        print ("2:", q)
        q += 10
        print ("2:", q)
        return q
    def f3(q):
        print ("3:", q)
        f1(q)
        print ("3:", q)
        q = f2(q)
        print ("3:", q)
    f3(3)
```

```
    def fun2(a,b):
        print("fun2:", a, b)
        a = a + b
        return a
    def fun3(a,b):
        print("fun3:", a, b)
        b = a * b
        return b
    a = 2
    b = 3
    fun2(a,b)
    c = fun2( fun3(a,a), fun3(b,b) )
    print ("*", c)
```

Write a function to determine if a time value is valid. The function will always be called with a single integer representing the time in HHMM format (e.g. 1245 would be interpreted as 12:45PM)

Your function should examine the supplied time value and determine whether (a) the time is valid and (b) whether it represents an AM or PM time.

The function should always return two values - a Boolean representing a the validity of the time value, and a string describing the time as either "AM", "PM" or "N/A" (in the case of invalidity)

Here are a few sample calls to the function and the expected output:

```
a,b = valid_time(1245)
print(a,b)                # True PM
a,b = valid_time(1259)
print(a,b)                # True PM
a,b = valid_time(1265)
print(a,b)                # False N/A
a,b = valid_time(745)
print(a,b)                # True AM
a,b = valid_time(1416)
print(a,b)                # True PM
a,b = valid_time(2850)
print(a,b)                # False N/A
a,b = valid_time(150)
print(a,b)                # True AM
```

Write your function to work exactly as described above, and document your function using IPO notation.

For this program you will be writing a FUNCTION to determine the greatest common divisor (GCD) shared between two integers. Your function should accept two integers as arguments and return the greatest common divisor shared between these two arguments. For the purpose of this question, you can always assume that the values being tested are integers (no data validation necessary). Also note that negative numbers should be allowed, since **`gcd(x, y)`** yields the same result at **`gcd(|x|, |y|)`**

Here's are a few examples of how your function should operate. *Your function should operate perfectly using the sample code below*. Do not add any additional features that are not represented here!

```
a1 = gcd(5,10)
print(a1)           # 5

a2 = gcd(10,5)
print(a2)           # 5

a2 = gcd(-10,5)
print(a2)           # 5

a2 = gcd(10,-5)
print(a2)           # 5

a3 = gcd(11,17)
print(a3)           # 1

a4 = gcd(58,24)
print(a4)           # 2

a4 = gcd(-58,-24)
print(a4)           # 2

a5 = gcd(18,12)
print(a5)           # 6
```

Note that you are only writing a FUNCTION for this question – you do not need to write a full program that prompts the user for input, etc. Ensure that your function will work perfectly when run using the sample code above. *Document your function using IPO Notation*.

Keep in mind that your program needs to work with every possible input (the integers being used above are simply examples – you cannot "hard code" your program to only use these values!)

Python Command Index

Core Language Elements and Functions	**Module Functions**
`and`	`random.randint`
`break`	`str.lower`
`continue`	`str.upper`
`def`	`time.time`
`elif`	
`else`	
`float`	
`for`	
`format`	
`if`	
`in`	
`import`	
`input`	
`int`	
`len`	
`not`	
`or`	
`print`	
`return`	
`str`	
`while`	

ASCII Code Table

0	<NUL>	32	<SPC>	64	@	96	`	128	Ä	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Å	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Ç	162	¢	194	¬	226	‚
3	<ETX>	35	#	67	C	99	c	131	É	163	£	195	√	227	„
4	<EOT>	36	$	68	D	100	d	132	Ñ	164	§	196	ƒ	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Ö	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	∆	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(	72	H	104	h	136	à	168	®	200	»	232	Ë
9	<TAB>	41	)	73	I	105	i	137	â	169	©	201	…	233	È
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	ã	171	´	203	À	235	Î
12	<FF>	44	,	76	L	108	l	140	å	172	¨	204	Ã	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Õ	237	Ì
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	Œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	–	240	
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	“	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	”	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	‘	244	Ù
21	<NAK>	53	5	85	U	117	u	149	ï	181	µ	213	’	245	ı
22	<SYN	54	6	86	V	118	v	150	ñ	182	∂	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	∑	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	∏	216	ÿ	248	¯
25	<EM>	57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	∫	218	⁄	250	˙
27	<ESC>	59	;	91	[	123	{	155	õ	187	ª	219	€	251	˚
28	<FS>	60	<	92	\	124	|	156	ú	188	º	220	‹	252	¸
29	<GS>	61	=	93	]	125	}	157	ù	189	Ω	221	›	253	˝
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	ﬁ	254	˛
31	<US>	63	?	95	_	127	<DEL>	159	ü	191	ø	223	ﬂ	255	ˇ