P	f1(P)	f2(P)	f3(P)=~P	f4(P)
0	1	0	1	0
1	1	0	0	1

P	Q	f1(P,Q)	f2(P,Q)	......
0	0	0	0	......
0	1	0	0	......
1	0	0	0	......
1	1	0	1	......

Don't need to worry about proving all of these as you only need to show that your set can generate {~, *, v, =>, <=>) because as per the slides this set is already adequte

So to prove . is not adequate you can show it doesn't genearate f3(P)=~P

P	P*P	P*P*P	P*P*P*P	.....	~P
0	0	0	0	0	1
1	1	1	1	1	0

no way to genete ~ with .*so {*} is no adequate (also can't generate f1(P), f2(P) above)

~ and * are given as they are in {~,*}

Use truth table to show other ones (maybe you can do it without truth tables but it is good to check)

Show we can generate v					Not unique
P	Q	PvQ	~P*~Q	~(~P*~Q)	~((~P*~Q).(~P*~Q))
0	0	0	1	0	0
0	1	1	0	1	1

1	0	1	0	1	1
1	1	1	0	1	1

Show we can genrate P=>Q

From textbook

P=>Q	=	~PvQ

Substitute the formula we already have for v

~PvQ	=	~(~(~P)*~Q)	=	~(P*~Q)

Don't need to check with truth table but I will do it so you can see

P	Q	P=>Q	P*~Q	~(P*~Q)
0	0	1	0	1
0	1	1	0	1
1	0	0	1	0
1	1	1	0	1

It works the other way too. If you have => you can convert to v using ~~P=P

Simialry from the text

P <=> Q	=	(P=>Q).(Q=>P)

Again substitute the formila we have for P=>Q from above

(P=>Q)*(Q=>P)	=	~(P*~Q)*~(~P*Q)

Then we are done as we have show we can write all standard connectives with {~,*}

(I will not do the truth table for this one)

So once you have ~, * , v you can use the substitution rules to get => and <=>