Task 1: minSup = 0.15, minConf = 0.8

Calculating support:

A = 4/5 = 0.8, **B** = 5/5 = 1, **C** = 4/5 = 0.8, **D** = 3/5 = 0.6, **E** = 1/5 = 0.2, **F** = 1/5 = 0.2

All pass minSup requirement, so now we create all sets of length-2 and recalculate support before pruning (sets marked red are removed, missing combinations have been pruned):

Set	Set-support	Set	Set-support
{A,B}	4/5 = 0.8	{B,E}	1/5 = 0.2
{A,C}	3/5 = 0.6	{B,F}	1/5 = 0.2
{A,D}	3/5 = 0.6	{C,D}	3/5 = 0.6
{A,E}	0	{C,E}	1/5 = 0.2
{A,F}	1/5 = 0.2	{C,F}	1/5 = 0.2
{B,C}	4/5 = 0.8	{D,E}	0
{B,D}	3/5 = 0.6	{D,F}	1/5 = 0.2
{E,F}	0		

Sets of Length-3:

Set	Set-Support	Set	Set-Support
{A,B,C}	3/5 = 0.6	{B,C,D}	3/5 = 0.6
{A,B,D}	3/5 = 0.6	{B,C,E}	1/5 = 0.2
{A,B,F}	1/5 = 0.2	{B,C,F}	1/5 = 0.2
{A,C,D}	3/5 = 0.6	{B,D,F}	1/5 = 0.2
{A,C,F}	1/5 = 0.2	{C,D,F}	1/5 = 0.2
{A,D,F}	1/5 = 0.2		

Sets of Length-4:

Set	Set-support	Set	Set-support
{A,B,C,D}	3/5 = 0.6	{B,C,D,F}	1/5 = 0.2
{A,B,C,F}	1/5 = 0.2	{A,C,D,F}	1/5 = 0.2
{A,B,D,F}	1/5 = 0.2		

Sets of Length-5:

Set	Set-support
{A,B,C,D,F}	1/5 = 0.2

Then calculating confidence for 4 of the 97 rules output by the algorithm:

Rule	Rule Confidence	Rule	Rule Confidence
{A} ➔ {B}	(AB/A) = 0.8/1 = 0.8	{A,B,C,D} ➔ {F}	(ABCDF/ABCD) = 0.2/0.6 = 0.34
{A,B} ➔ {C}	(ABC/AB) = 0.6/0.8 = 0.75	{A,D} ➔ {C}	(ADC/AD) = 0.6/0.6 = 1

for all non-empty subset s of l where l is a frequent itemset, take s ➔ (l – s), giving:

{F} ➔ {ABDC}, {DF} ➔ {ABC}, {CF} ➔ {ABD}, {BF}➔{ADC}, {BDF}➔{AC}, {BCF}➔{AD}, {AF}➔{BDC}, {ACF}➔{BD}, {ABF}➔{DC}, {ABCF}➔{D}, {DCF}➔{AB}, {BDCF}➔{A}, {ADF}➔{BC}, {ABDF}➔{C}, {ADCF}➔{B}, {F}➔{BDC}, {F}➔{ADC}, {F}➔{ABD}, {F}➔{ABC},{DF}➔{AC}, {D}➔{ABC}, {CF}➔{BD}, {CF}➔{AD}, {BF}➔{DC}, {BF}➔{AD}, {BF}➔{AC}, {BD}➔{AC}, {BCF}➔{D}, {AF}➔{DC}, {AF}➔{BD}, {ACF}➔{D}, {AC}➔{BD}, {ABF}➔{D}, {ABC}➔{D}, {DC}➔{AB}, {BDC}➔{A}, {AD}➔{BC}, {ABD}➔{C},

{DF}➔{BC},{DF}➔{AB}, {DCF}➔{A}, {CF}➔{AB}, {BDF}➔{C}, {BDF}➔{A}, {BCF}➔{A}, {AF}➔{BC}, {ADF}➔{C}, {ABF}➔{C}, {DCF}➔{B}, {ADF}➔{B}, {ADC}➔{B}, {ACF}➔{B}, {F}➔{DC}, {F}➔{BD}, {F}➔{AD}, {F}➔{AC}, {D}➔{AC}, {CF}➔{D}, {BF}➔{D}, {AF}➔{D}, {AC}➔{D}, {DC}➔{A}, {D}➔{BC}, {D}➔{AB}, {BD}➔{C}, {BD}➔{A}, {AD}➔{C}, {F}➔{BC}, {F}➔{AB}, {EB}➔{C}, {E}➔{BC}, {DF}➔{C}, {DF}➔{A}, {CF}➔{A}, {BF}➔{C}, {BF}➔{A}, {AF}➔{C}, {EC}➔{B}, {DF}➔{B}, {DC}➔{B}, {CF}➔{B}, {AF}➔{B}, {AD}➔{B}, {AC}➔{B}, {F}➔{D}, {D}➔{C}, {D}➔{A}, {F}➔{C}, {F}➔{A}, {E}➔{C}, {F}➔{B}, {E}➔{B}, {D}➔{B}, {C}➔{B}, {A}➔{B}, {B}➔{C}, {B}➔{A}

Task 3: no room on report but running code with minSup=0.15 & minConf=0.8 will produce the above results.

Task 4:
apriori(minsup = 0.1, minconf = 0.8, minlift = 1, 'supermarket.csv'):
Total Rules = 8066, Frequent Itemsets = 10282
{Total = high, canned fruit} ➔{Biscuits} (lift = 1.47, conf = 0.83) &
{Total = high, fruit, beef} ➔ {vegetables, bread&cake} (lift = 1.65, conf = 0.82)
With a low Support constraint, the rules returned have very high lift values compared to those where the support constraint was higher. With high lift, we can infer that the association between the antecedents & consequents in these rules is high as lift takes into account the popularity of the consequent & antecedent, instead of confidence which accounts for the antecedent popularity only.
apriori(minsup = 0.3, minconf = 0.8, minlift = 1, 'supermarket.csv'):
Total Rules = 17, Frequent Itemsets = 114
{Milk, Biscuits} ➔ {Bread&Cake} (lift = 1.17, conf = 0.84) &
{Margerine} ➔ {Bread&Cake} (lift = 1.11, conf = 0.8)
Given the heavy minSup constraint, items appearing in these rules appeared frequently across all transactions. Interesting because despite these items being crucial for each-other practically, their lift describes lower overall association for both antecedents with bread&cake across the dataset.
apriori(minsup = 0.3, minconf = 0.6, minlift = 1, 'supermarket.csv'):
Total Rules = 147, Frequent Itemsets = 114
With a low confidence constraint we see 2 interesting things in the rules generated: some items clearly hold low association {'party snack foods'}➔{'milk-cream'}(lift = 1.04, conf = 0.67) and the measures prove it, but also lift for the larger rules increased {'vegetables', 'biscuits'} ➔ {'fruit'}(lift = 1.24, conf = 0.8), producing rules that made sense.

apriori(minsup = 0.2, minconf = 0.8, minlift = 1.2, 'supermarket.csv'):
Total Rules = 51, Frequent Itemsets = 640
{'total = high', 'fruit', 'bread and cake'} ➔ {'vegetables'}(lift = 1.33, conf = 0.85) &
{'vegetables', 'milk-cream', 'frozen foods'} ➔ {'fruit'}(lift = 1.25, conf = 0.80)
I set minSup = 0.2 to produce more frequent itemsets,and wanted to see what would happen when lift was constrained higher. With these settings, while not only getting some rules with greater lift (therefore higher overall association), the confidence values were split between being very close to 0.8, or pushing quite a bit higher to 0.88-0.89 area. Those with the lower conf values but similar lift would've antecedents that occurred more than consequents, so when lift was calculated, it balanced the popularities out bringing higher lifts.

Task 5: By implementing a MinRelativeSup constraint, if MinRelSup is set to the same value as MinSup, the number of rules and itemsets generated will remain unchaged. However, if MinRelSup is set to a higher value (to compensate for division of 0.X by 0.Y numbers boosting the resultant value), it further restricts and reduces the number of rules & itemsets.

As a toy example, take these itemsets and support values from Task 1 and set MinRelSup = 0.6:

MaxSupport for 2-itemsets = Support({A,B}) = 0.8

- Support({A,B,C}) = 0.6, RelSup({A,B,C}) = itemsetSupport / maxSupport for k-1items = 0.6/0.8 = 0.75. RelSup({A,B,C}) > MinRelSup, therefore {A,B,C} is kept.
- Support({A,B,F}) = 0.2, RelSup({A,B,F}) = 0.2/0.8 = 0.25. RelSup({A,B,F}) < MinRelSup,

Therefore {A,B,F} and all supersets of {A,B,F} are discarded.

MaxSupport for 1-itemsets = Support({B}) = 1:

- Support({A,B}) = 0.8, RelSup({A,B}) = 0.8/1 = 0.8
- Support({B,D}) = 0.6, RelSup({B,D}) = 0.6/1 = 0.6

Both have RelSup > MinRelSup, therefore both are kept.

Implementation of MinRelSup still maintains the anti-monotonicity property of Apriori. This can be modelled as: ($\forall_{x,y}$ | x is a superset & contains subset y), $\text{Support}(x) < \text{Support}(y)$ & $\text{RelSup}(x) < \text{RelSup}(y)$.

Take {A,B,C} and {A,B} as x and y respectively;

- $\text{Support}(\{A,B\}) = 0.8$, $\text{Support}(\{A,B,C\}) = 0.6$, $\text{Support}(\{A,B\}) > \text{Support}(\{A,B,C\})$
- $\text{RelSup}(\{A,B\}) = 0.8$, $\text{RelSup}(\{A,B,C\}) = 0.75$, $\text{RelSup}(\{A,B\}) > \text{RelSup}(\{A,B,C\})$

The Anti-Monotone property essentially states that for any itemset with k items, its subsets will always have higher support: Support(k-itemset) will never be larger than its subsets. Therefore, by the example above, Anti-Monotonicity still holds with MinRelSup implementation.