Question 1: Shorties

a. True
From $A \leq_m B$, we know there exists function f s.t. $w \in A \iff f(w) \in B$,
From $\overline{B} \leq_m \overline{C}$, we know there exists function g s.t. $w \in \overline{A} \iff g(w) \in \overline{B}$,
Given $a \in A$, then $f(a) \in B$. It follows that $f(a) \notin \overline{B}$, so $g(f(a)) \notin \overline{C}$, so $g(f(a)) \in C$. Similarly, we have if $a \notin A$, then $f(a) \notin B$. It follows that $f(a) \in \overline{B}$, so $g(f(a)) \in \overline{C}$, so $g(f(a)) \notin C$

Therefore we have shown $a \in A \iff g(f(a)) \in C$ i.e. $A \leq_M C$

b. False
To prove A is map reducible to B, we need to show that there exists a function f such that $w \in A \iff f(w) \in B$. However, we don't require f to be onto or one-to-one. Therefore, we are not guaranteed a map g such that $a \in B \iff g(a) \in A$. So we can't guarantee that $B \leq_m A$

c. False
$B = \Sigma^*$ is clearly context free. Every language will be a subset of B but not all subset of B is decidable.

d. True
Since A context free, there exists a PDA P that accept A. We could then make a TM M that simulate P. Then M will recognise A. So A is recognisable Both A and $\overline{A}$ is recognisable, therefore, A is decidable.

Question 2: Context Free

$S \to SS|aSaSb|aSbSa|bSaSa|\epsilon$

We first prove that all strings generated by this production rule will be in C
We do this by induction:
Base case: $S \xrightarrow{1} x$, the only possible x is ϵ, thus $x \in C$
Induction Hypothesis: Assume $S \xrightarrow{i} x$ implies $x \in C$ for all $i \leq k$
Induction Step: Suppose now $S \xrightarrow{k+1} y$,

case(1): $S \xrightarrow{1} SS \xrightarrow{k} y$. Then, we know that y = y_1y_2 where y_1, y_2 are both S after k steps of production. By induction hypothesis, $y_1, y_2 \in C$. Thus $\#a(y) = \#a(y_1) + \#a(y_2) = 2\#b(y_1) + 2\#b(y_2) = 2\#b(y)$. Thus, $y \in C$

case(2):$S \xrightarrow{1} aSaSb \xrightarrow{k} y$ (similar case for $S \to aSbSa$ and $S \to bSaSa$). Then y = ay_1ay_2b where y_1, y_2 are both S after k steps of production. By induction hypothesis, $y_1, y_2 \in C$.
Thus $\#a(y) = 1 + \#a(y_1) + 1 + \#a(y_2) = 2\#b(y_1) + 2\#b(y_2) + 2 = 2\#b(y)$.
Thus, $y \in C$

case(3) $S \xrightarrow{1} \epsilon$. S gets to a terminal in one step so it's not within our discussion here. Also $\epsilon \in C$

Thus we have the strings produced by our grammar is always in C.

Now we want to show that if $x \in C$, then $S \xrightarrow{*} x$
Induction on length of x:
Base case: $|x| \leq 3$, $x = \epsilon$ or aab or aba or baa
IH: For all $x \in C$, $|x| \leq k$, $S \xrightarrow{*} x$
IS: Let $d(x) = \#a(x) - 2\#b(x)$.

Case(1): d(x) = 0 for some proper prefix of x. Then write x= yz, we know that $y \in C, z \in C$.
By induction $S \xrightarrow{*} y$, $S \xrightarrow{*} z$.
Thus $S \to SS \xrightarrow{*} yS \xrightarrow{*} yz = x$

Case(2) $d(y) > 0$ for all proper prefix y of x.
Then there will be many ways we can write, x = ayazb. And in at least one way we have $y, z \in C$. (Proof of this at the bottom of the page)
By IH, By induction $S \xrightarrow{*} y$, $S \xrightarrow{*} z$.
Thus $S \to aSaSb \xrightarrow{*} ayaSb \xrightarrow{*} ayazb = x$

Case(3) $d(y) < 0$ for all proper prefix y of x.
Then there will be many ways we can write, x = byaza. And in at least one way we have $y, z \in C$.(similar reasoning as the one above)
By IH, By induction $S \xrightarrow{*} y$, $S \xrightarrow{*} z$.
Thus $S \to bSaSa \xrightarrow{*} byaSa \xrightarrow{*} byaza = x$

Case(4) $d(y) < 0$ for some proper prefix y of x and $d(x) < 0$ for some proper prefix of x and $d(x) \neq 0$ or all proper prefix of x (For example, aaabba)
Then there will be many ways we can write, x = aybza. And in at least one way we have $y, z \in C$.(similar reasoning as the one above)
By IH, By induction $S \xrightarrow{*} y$, $S \xrightarrow{*} z$.
Thus $S \to aSbSa \xrightarrow{*} aybSa \xrightarrow{*} aybza = x$

Thus by induction hypothesis, if $x \in C$, then $S \xrightarrow{*} x$

In conclusion, we have shown our grammar generates C.

Claim: $x \in C$, $d(y) > 0$ for all proper prefix y of x and x = amanb. Then in at least one way we have $m, n \in C$.
Proof:Since $d(y) > 0$ for all proper prefix y of x, first 3 characters of x can't be b, so x = aaawb for some w. Then let $m = \epsilon$, $n = aw$ will suffice.

Question 3: Not Context Free

$D = \{x\#y | x, y \in \{a, b\}^*, |x| = |y|, \#a(x) = \#a(y)\}$

Assume D is a context free language.
Then pumping lemma should apply to D.
Given a pumping length $p > 0$.
let $w = a^p b^p \# a^p b^p$
we can then write w = uvxyz where $|vy| > 0, |vxy| \leq p$, then either:
1) z contains #. Then by pumping w zero times, we get $w' = uxz$. since # is in z, we know that characters after # will be unchanged and there is less character in the first half of the string. So we no longer have equal number of characters before and after #, so $w' \notin D$

2) u contains #. Then it is similar to case 1). If we pump w 0 times, there $w' = uxz$will have character before # than after #. so $w' \notin D$.

3) x contains #. We want same number of characters before # and after # after pumping, so we can assume that $|v| = |y| > 0$. Since $|vxy| < p$, we know that v won't contain any a and y will contain at least one a. That means if we pump w 0 times, strings after # will have less a . So it is not in D after pumping twice

4) v or y contains #. If we pump w 0 times, we will have no # in the string. So it is not in D after pumping.

Thus, we have shown that after pumping 0 times, $w' \notin D$. So our original assumption(D is context free language) is False. So D is not context free language.

Question 4: But it is Decidable

M = on input string w,

1. Scan though the tape from left to right.

If it sees first b after #, then go to reject state.

If it sees first the first b before #, then change that b to $\overline{b}$ and move the head to the #. Then scan through the rest of the tape until it sees first b (or reach the end). If it sees b , then change that b to $\overline{b}$ and move the head to the start of the tape and repeat step 1. If it does not see b, then go to reject state.

If it does not see any b, move its head to the start of the tape and then go to step 2.

2. Scan though the tape from left to right.

If it sees first a after #, then go to reject state.

If it sees first the first a before #, then change that a to $\overline{a}$ and move the head to the #. Then scan through the rest of the tape until it sees first a (or reach the end). If it sees a, then change that a to $\overline{a}$ and move the head to the start of the tape and repeat step 2. If it does not see a, then go to reject state.

If it does not see any a, then go to accept state.

Explanation of my TM: In step 1, we check if there are equal number of b's on both side of #.In step 2, we check if there are equal number of a's on both side of #. If either is false, then we rejects.

Question 5: Overlapping TM's

$F = \{\langle M_1, M_2\rangle | M_1, M_2 \text{ are TM's and } L(M_1) \cap L(M_2) \neq \emptyset\}$

First, let's show F is recognisable:
Pick an ordering on strings x1, x2, ... Start with i = 1, simulate both M_1, M_2 on strings x1 through xi for i steps. If any of these is accepted by both M_1 and M_2, accept. Otherwise, increment i by 1 and continue. We have built a machine that recognise F, thus F is recognisable

We now show that F is not decidable:
We will prove $A_{TM} \leq_M F$

$f =$ "on input $\langle M, x\rangle$:

1. compute/construct a TM M_1, M_2 such that $\langle M, x\rangle \in \mathsf{A}_{TM} \iff \langle M_1, M_2\rangle \in \mathsf{F}$:

   $M_1 =$" on input y:
   1. Ignore / erase y.
   2. Run M on x, accept y if and only if M accepts x. "

   $M_2 =$" on input y:
   1. Ignore / erase y.
   2. Run M on x, accept y if and only if M accepts x. "

2. Return $\langle M_1, M_2\rangle$. "

We claim that M halts on x if and only if $\langle M_1, M_2\rangle$ is in F.

$$\begin{aligned}
\langle M, x\rangle \in A_{TM} &\Rightarrow M \text{ halts on } x \\
&\Rightarrow M_1, M_2 \text{ accepts all input } y \\
&\Rightarrow L(M_1) \cap L(M_2) \neq \emptyset \\
&\Rightarrow \langle M_1, M_2\rangle \in F \\
&\Rightarrow f(\langle M, x\rangle) \in F
\end{aligned}$$

$$\begin{aligned}
\langle M, x\rangle \notin A_{TM} &\Rightarrow M \text{ does not halt on } x \\
&\Rightarrow M_1, M_2 \text{ accepts no string} \\
&\Rightarrow L(M_1) \cap L(M_2) = \emptyset \\
&\Rightarrow \langle M_1, M_2\rangle \notin F \\
&\Rightarrow f(\langle M, x\rangle) \notin F
\end{aligned}$$

Thus, we have shown that $A_{TM} \leq_M F$. Since A_{TM} is not Turing decidable, we know that F is also not decidable