Problem 1

$\mathcal{M}$ represents the matching at each iteration. The order of execution is as follows:

- **Round 1:**
  - h_3 proposes to s_2; s_2 accepts.
  - $\mathcal{M} = [(h_3, s_2)]$
- **Round 2:**
  - h_2 proposes to s_2; s_2 declines.
  - $\mathcal{M} = [(h_3, s_2)]$
- **Round 3:**
  - h_2 proposes to s_1; s_1 accepts.
  - $\mathcal{M} = [(h_3, s_2); (h_2, s_1)]$
- **Round 4:**
  - h_1 proposes to s_1; s_1 declines.
  - $\mathcal{M} = [(h_3, s_2); (h_2, s_1)]$
- **Round 5:**
  - h_1 proposes to s_2; s_2 accepts, rejecting h_3.
  - $\mathcal{M} = [(h_1, s_2); (h_2, s_1)]$
- **Round 6:**
  - h_3 proposes to s_1; s_1 declines.
  - $\mathcal{M} = [(h_1, s_2); (h_2, s_1)]$
- **Round 7:**
  - h_3 proposes to s_3; s_3 accepts.
  - $\mathcal{M} = [(h_1, s_2); (h_2, s_1); (h_3, s_3)]$

Problem 2

(a) Consider input with n hospitals $h_1, h_2, \ldots, h_n$ and n students $s_1, s_2, \ldots, s_n$ with the following preferences:

- All hospitals have the same preferences for students: $s_1 > s_2 > \cdots > s_n$.
- All students have the same preferences of hospitals: $h_n > h_{n-1} > \cdots > h_1$.

For $n = 3$, we will have 3 hospitals and 3 students with preferences as follows:

$$h_1 : s_1 > s_2 > s_3$$

$$h_2 : s_1 > s_2 > s_3$$

$$h_3 : s_1 > s_2 > s_3$$

$$s_1 : h_3 > h_2 > h_1$$

$$s_2 : h_3 > h_2 > h_1$$

$$s_3 : h_3 > h_2 > h_1$$

We now show that with the above preferences, there is an ordering of the "free" hospitals that forces the Gale-Shapley algorithm to execute $\Omega(n^2)$ iterations. Suppose that the hospitals are all placed in a queue in the order $h_1, h_2, \ldots, h_n$ initially. The algorithm picks hospitals from the front of the queue, whereas hospitals who become free during the course of the algorithm join at the back of the queue. Using this ordering on "free" hospitals, we see that the algorithm executes as follows:

(1) h_1 proposes to s_1 and this is accepted.

(2) h_2 proposes to s_1 and h_1 is kicked out and h_2 gets matched with s_1.

(3) h_3 proposes to s_1 and h_2 is kicked out and h_3 gets matched with s_1. ·
·
·

(i) h_i proposes to s_1 and h_{i-1} is kicked out and h_i gets matched with s_1. ·
·
·

(n) h_n proposes to s_1 and h_{n-1} is kicked out and h_n gets matched with s_1.

At the end of this sequence of n proposals, h_n is matched with s_1 and $h_1, h_2, h_3, \ldots, h_{n-1}$ are all free. Then a sequence of $n - 1$ proposals are made to s_2 (by hospitals in the order $h_1, h_2, \ldots, h_{n-1}$) at the end of which h_{n-1} is matched with s_2 and $h_1, h_2, h_3, \cdots, h_{n-2}$ are free.

Continuing this manner, we see that the total number of proposals made by the algorithm are:

$$n + (n-1) + (n-2) + \cdots + 2 + 1 = \frac{n(n+1)}{2}.$$

This shows that the worst case number of iterations of the while loop in Gale-Shapley algorithm for this input is $\Omega(n^2)$.

(b) Consider input with n hospitals $h_1, h_2, \ldots, h_n$ and n students $s_1, s_2, \ldots, s_n$ with the following preferences:

- Each hospital h_i prefers student s_i over the rest: $s_i > \cdots$.

– Each student s_i prefers hospital h_i over the rest: $h_i > \cdots$.

For $n = 3$, we will have 3 hospitals and 3 students with preferences as follows:

$$h_1 : s_1 > s_2 > s_3$$

$$h_2 : s_2 > s_3 > s_1$$

$$h_3 : s_3 > s_1 > s_2$$

$$s_1 : h_1 > h_2 > h_3$$

$$s_2 : h_2 > h_3 > h_1$$

$$s_3 : h_3 > h_1 > h_2$$

Now suppose that the hospitals are placed in a queue in the order $h_1, h_2, \cdots, h_n$ initially. Using this ordering, the execution of the algorithm is as follows:

(1) h_1 proposes to s_1 and this is accepted.

(2) h_2 proposes to s_2 and this is accepted.

(3) h_3 proposes to s_3 and this is accepted.

.

.

.

(n) h_n proposes to s_n and this is accepted.

At the end, each hospital h_i is matched with student s_i. It takes $O(n)$ iterations of the while loop in Gale-Shapley algorithm to complete for this input.

Problem 3

(a) The given preferences are:

- s_1 prefers m_1 over any m_i
- m_1 prefers s_1 over any s_i

Assume that the matching $\mathcal{M}$ produced by the Gale-Shapley algorithm does not include the (s_1, m_1) pair.

Therefore, s_1 must be matched with any m_i, and m_1 must by matched with any s_i, as shown:

$$(s_1, m_i), (s_i, m_1)$$

Given that the Gale-Shapley algorithm produces stable teams, at least one of the following statements must be true:

- s_1 prefers any m_i over m_1
- m_1 prefers any s_i over s_1

Otherwise, the matchings would be unstable, and members of each pair would want to switch with each other.

However, the given preferences contradict the preferences produced by the assumption. Therefore, the matching $\mathcal{M}$ produced by the Gale-Shapley algorithm must include the (s_1, m_1) pair.

(b) The given preferences are:

- s_1 prefers m_1 over any m_i
- m_1 prefers s_1 over any s_i

By the definition of unstable matching, there must exist a matched pair (s_i, m_i) and an alternative pair (s_j, m_j) in which the following preferences are true:

- m_i: $s_j > s_i$
- s_j: $m_i > m_j$

Or

- m_j: $s_i > s_j$
- s_i: $m_j > m_i$

With unstable pairs, each school and student would prefer the partner of the other pair.

Given that m_1 and s_1 are not in the same pair, the following pairs will exist in any matching without the $(s_1,\ m_1)$ pair:

$$(s_1, m_i), (s_j, m_1)$$

With the given preferences, we know that m_1 and s_1 paired with any other school and any other student will be unstable, as s_1 and m_1 would prefer to switch pairs. This aligns with the preferences outlined by the definition of an unstable matching. Therefore, any matching without s_1 and m_1 would be unstable.

(c) The given preferences are:

Problem 3 continued on next page. . .

– s_1 prefers any m_i over m_1
– m_1 prefers any s_i over s_1

Assume there exists a matching $\mathcal{S}$ with pair (s_1, m_1) that is unstable.

Consider input with n schools $m_1, m_2, \ldots, m_n$ and n students $s_1, s_2, \ldots, s_n$ with the following preferences:

– Each school m_i prefers student s_n over the rest: $s_n > \cdots > s_1$.
– Each student s_i prefers school m_n over the rest: $m_n > \cdots > m_1$.

The Gale-Shapley algorithm will end with the following pairs:

$$(s_n, m_n), \cdots (s_1, m_1).$$

By the definition of an unstable pair, since (s_1, m_1) is unstable, there must exist an alternative pair, (s_i, m_i), in which the following is true:

– s_i prefers m_1 over the m_i it is matched with
– m_i prefers s_1 over the s_i it is matched with

However, if no student prefers m_1 and no school prefers s_1, there does not exist such a pair.

By contradiction, the statement that "any matching which pairs m_1 and s_1 is unstable" is false.

Problem 4

(a) One example is the following:

Initialize: Set F representing free players and set T representing players already in a team.
while $F \neq \emptyset$ **do**
  Pick a player A from F.
  A proposes to their first two preferences B and C.
  if B and C are free **then**
    They form a team and join T.
  end if
  if B or C are already in a team **then**
    if B and C prefer the proposed team to their current team **then**
      They switch teams and their previous team disbands.
      Their previous teammates return to F.
    else
      A proposes to their next 2 preferences.
    end if
  end if
end while

(b) The input can be defined as the following:

$$A : B > C > D > E > F > G > H > I$$

$$B : C > D > E > F > G > H > I > A$$

$$C : D > E > F > G > H > I > A > B$$

$$D : E > F > G > H > I > A > B > C$$

$$E : F > G > H > I > A > B > C > D$$

$$F : G > H > I > A > B > C > D > E$$

$$G : H > I > A > B > C > D > E > F$$

$$H : I > A > B > C > D > E > F > G$$

$$I : A > B > C > D > E > F > G > H$$

With respect to the preferences and part (a) the matching will be $(A, B, C), (D, E, F), (G, H, I)$ and the unstable team would be (C, F, I).