Homework2

Qinglin Li, 5110309074

Problem 1

Protocol:

Round 1

The sender sends message m to everybody

Round 2—Round $f + 2$

The sender and all processes that have received the message m propose for m, others propose for `SF`

Run the consensus protocol with the decision function $f(V) = \begin{cases} m & (m \in V) \\ SF & (m\ V) \end{cases}$

Proof:

Termination

Since the consensus protocol terminates, this protocol also terminates.

Agreement

If a correct process have delivered m, it must have decided m in the consensus protocol.
According to the agreement of the consensus protocol, all correct processes must have decided m. So all correct processes have delivered m.

Validity

According to the validity of the consensus protocol, if the sender is correct, all correct processes should eventually have decided m and delivered m;

Integrity

According to the consensus protocol, every correct process delivers at most one message.
If a correct process have delivered m, it must have decided m in the consensus protocol. According to the integrity of the consensus protocol, there must be some process have proposed m which is sent form the sender.

Problem 2

Protocol:

Round 1

The sender sends message m to everybody

Round 2—Round $f+1$

The sender sends message m to everybody.
The sender and all processes that have received the message m propose for m, others propose for `SF`
Run the consensus protocol for f round as if there are at most $f-1$ processes crash with the decision function $f(V) = \begin{cases} m & (m \in V) \\ SF & (m\ V) \end{cases}$

Proof:

Termination

According to the termination of the consensus protocol, this protocol must terminate.

Agreement

If the sender hadn't crashed until round 2, every correct process should have received m. So every correct process should decide m and then deliver m.
If the sender crashed before the second round, there are at most $f-1$ faults remaining. So the sonsensus protocol can ensure agreement.

Validity

If the sender is correct, all correct processes must have received the message m from the sender in the second round. So all of them deliver m.

Integrity

The decision function f ensures every correct process delivers at most one 1 message. If it delivers $m \neq SF$, then sender must have broadcast m

Problem 3

1. **Protocol:**

   Suppose sender sends the message m and p is an arbitrary process.
   In round $k(1 \leq k \leq t+1)$:

   (a) If p is the sender or p has received m in the previous rounds, p sends m to all.

   (b) p receives messages sent in round k.

   In round $t+2$:

   (a) If p is sender or p has received m in the previous rounds,p sends m to all. Otherwise, p sends SF to all.

   (b) p receives messages sent in round $t+2$.

   (c) If p have received some message m for more than $\frac{n}{2}$ times, p delivers m.

Proof:

Termination

Since the first $t+1$ rounds are same as the TRB protocol except for message dilivering, all correct processes should send the same message in round $t+2$ and then they eventually deliver the message.

Uniform Agreement

If a process p delivered $m \neq SF$, it must have received m for more than $\frac{n}{2}$ times in round $t+2$. So more than $\frac{n}{2}$ processes sent m in round $t+2$ and they must have received m before round $t+2$.
Since the first $t+1$ rounds are same as the TRB protocol except for message dilivering, by the end of round t + 1, all correct processes should have received m.
So all correct processes should deliver m in round $t+2$.

If a process p delivered $m = SF$, it must have received SF for more than $\frac{n}{2}$ times in round $t+2$. So more than $\frac{n}{2}$ processes sent SF in round $t+2$ and they must haven't received m before round $t+2$.
Since the first $t+1$ rounds are same as the TRB protocol except for message dilivering, by the end of round t + 1, all correct processes should have received m if any correct process have reveived m.
So none of the correct process have ever received m and all correct processes should deliver SF in round $t+2$.

Validity

If the sender is correct, all correct processes should have received m.
So in round $t+2$ every correct process should have received m for more than $\frac{n}{2}$ times because $n > 2t$.
All correct processes delivered m

Integrity

In round t + 2, every correct process should deliver at most one message.
If a correct process delivered m, then the sender must have sent m.

2. **Proof:**

   (By contradiction)
   Suppose there is an protocol with $n \leq 2t$
   The processes can be divided to two groups X and Y whose size are less than or equal to t. WLOG, let the sender in X.
   If all processes in X crashed at the beginning, all the processes in Y should deliver SF.
   But if all processes in X are correct and all process in Y cannot receive messages. Because these two situations are similar for every process in Y, they should deliver SF. However, every process in X should deliver m.
   Contradiction to **uniform agreement**.

Problem 4

Termination

Since the vector V_p have at least one element that is not $\perp$, every process should eventually decide.

Agreement

According to Lemma 2, for every correct process p, V_p are the same in the end. So the decisions must be the same.

Integrity

At the end of the protocol, every correct process should decide only one value.
If the value v is decided, there must be some process p having v as element of V_p by definition of the operations.

Validity

If all process propose for v, the only possible value except for $\perp$ in every V_p at the end is v.
So every correct process should eventually decide v.

Problem 5

Suppose $n = 3$ and $f = 1$

	p_1	p_2	p_3
input	0	1	1
sent a-value	0	1	1
received a-values	$(0,1)$	$(1,1)$	$(1,1)$
sent b-value	$\perp$	1	1
received b-value	$(1,\perp)$	$(1,\perp)$	$(1,1)$
decision	none	none	1

Suppose then p_1 and p_2 have 0 as their a-value and p_3 have a really large latency so that p_1 and p_2 always ignore its messages.
So p_1 and p_2 would decide 0
Agreement is broken.