Q1：

```
cookie=0x0,  duration=104.580s,  table=0,  n_packets=660960,  n_bytes=27761060,
priority=0 actions=CONTROLLER:65535
```

Q2：

The purpose of this initial flow entry is to handle packets that do not match any existing flow entries. It is the ’table-miss’ flow entry, which ensures that such packets are forwarded to the controller. The controller then decides whether to create new flow entries based on the characteristics of the incoming packet, and instructs the switch to install those entries.

Q3：

```
# Install flow to avoid future packet-in for this pair
      if out_port != ofproto.OFPP_FLOOD:
            if eth.ethertype == ether_types.ETH_TYPE_IP:
                  ip = pkt.get_protocol(ipv4.ipv4)
                  srcip = ip.src
                  dstip = ip.dst
                  protocol = ip.proto

                  # Define match rule for ICMP protocol
                  if protocol == in_proto.IPPROTO_ICMP:
     match = parser.OFPMatch(eth_type=ether_types.ETH_TYPE_IP, in_port=in_port,
ipv4_src=srcip, ipv4_dst=dstip, ip_proto=protocol)
```

Q4：Flooding attacks force the switch to install new flow entries for each unique source-destination pair by constantly sending traffic with different source and destination addresses. As a result, the switch has to create a new flow entry for each unique pair, quickly exhausting the flow table. Once the flow table is full, the switch can no longer install new flow entries, causing it to drop packets or send an error message to the controller (OFPFMFC_TABLE_FULL), effectively leading to a Denial of Service (DoS) condition.