Which of the following are reasons why there has been growing interest in programmable data planes?



There are not enough features in the OpenFlow specification.



**The OpenFlow specification provides no easy way to modify packet format.**

****

**Emerging chipsets such as RMT and FlexPipe make it possible to consider a richer set of match/action primitives.**

****

**The original OpenFlow protocol was not designed "from scratch", but rather was constrained by the hardware of existing switches.**

****

**Adding new features to the data plane requires changes to both forwarding elements and controllers.**

1 point

2.Question 2

What are some of the desired features in future SDN switches that motivate a programmable data plane?



**A configurable packet parser that is not tied to a specific header format.**



Processing packets at higher rates.



**Flexible match-action tables, such as the ability to compose multiple match-action tables in sequence or in parallel.**

****

**A wider range of packet processing primitives.**



Reducing energy consumption for custom packet processing.

1 point

3.Question 3

Which of the following best describes protocol independence?



The ability to support multiple protocols running in parallel on the same hardware.



The ability to run a network with both legacy OpenFlow switches and switches that support custom packet processing.



**The ability to configure a packet parser in hardware that is agnostic to (and not constrained by) the control protocol.**

****

The ability to evolve an existing control protocol without breaking interoperability.



The ability to run a switch independently from the controller.

1 point

4.Question 4

Which of the following best describes the notion of target independence?



Being able to port a "compiled" configuration from one target to another without modification.



**Relying on a compiler to translate a high-level specification of packet processing to configuration that will run on the target switch**.



Assuring correct switch operation regardless of the target destination of a packet.



Having a guarantee that the program will always run, regardless of the constraints of the hardware.



**Being able to write a packet processing program without knowledge of the underlying switch details.**

1 point

5.Question 5

Which of the following describe functions of the P4 parser?



**Verifying that the description of packet parsing is consistent with the behavior of an existing hardware switch (e.g., a legacy OpenFlow switch).**



Translating rules to physical tables.



**Translating a packet processing specification to a state machine.**



Mapping logical tables to underlying switch resources.



Constructing a graph of dependencies between different logical tables.

1 point

6.Question 6

Which of the following best describe the relationship between a language like P4, an assembler like NetASM, and target programmable hardware (e.g., RMT)?



P4 has verifiable semantics, whereas NetASM does not.



A language like NetASM would "compile" to an assembly language like P4, which would then specify a one-to-one relationship with resources on the target forwarding hardware.



**A language like P4 would "compile" to an assembly language like NetASM, which would then specify a one-to-one relationship with resources on the target forwarding hardware.**



NetASM and P4 are two examples of high-level programming languages; they are interchangeable.



P4 is much more amenable to "cost semantics" (i.e., figuring out how much a particular operation would cost in terms of power or area) than NetASM.

1 point

7.Question 7

Which of the folloiwing instruction types create state elements in NetASM?



DRP



LBL



**MKT**

****

**MKR**



JMP

1 point

8.Question 8

Which of the following are true about locally contained applications?



Locally contained applications forward packets more slowly than their stateless counterparts.



**Locally contained applications allow the hardware pipeline to make switch-local state modifications.**

****

**A MAC learning switch can be implemented as a locally contained application.**



Locally contained applications can be implemented with regular OpenFlow instructions.



**A locally contained application can implement SDN logic without ever requiring a packet to be sent to the controller.**

1 point

9.Question 9

Which of the following is true about the stateful MAC learner described in the lesson?



**The match table outputs an index corresponding to the row in the table with a matching destination MAC address.**



The match table outputs an index corresponding to the output port for the packet.



**The register keeps track of how many table entries are in the match and modify tables.**

****

**The modify table takes as input an index and modifies the packet's outport header to the corresponding value from the modify table**.



The modify table modifies the switch local state that maps a destination MAC address to an output port.

1 point

10.Question 10

What are some examples of optimizations that a compiler could perform on NetASM instructions?



**Reordering instructions so that subsequent tables need not be as large (e.g., not storing state for unmatched packets).**



Preventing bufferbloat for real-time application traffic.



Forwarding packets for high-volume flows at higher rates.



**Composing or decomposing tables to save either on table size or table width.**

****

**Only instantiating hardware resources (e.g., table entries) for flows that actually have nonzero traffic volume.**

1 point