## Question 1

**Question Type: MultipleChoice**

You need to build a custom module to add a notice to the Advanced Inventory section on the product form. The notice text can be specified in the system configuration. How do you implement this?

**Options:**

**A-** You add a plugin on the form \Magento\Cataiog\Ui\DataProvidr\Product\Form\ProductDataProvider to add a field in the Advanced Inventory section

**B-** YOU add a plugin on Magento\CatalogInventory\Ui\DataProvider\Product\Form\Modifier\AdvancedInventory to modify the Section metadata

**C-** You inject a custom block into the reference container in the catalog_product_index layout

**D-** You add a field to product_form. xml as a child of the stock_data fieldset

**Answer:**

D

## Question 2

**Question Type:** **MultipleChoice**

You are working on a project where many catalog managers often change products in the admin panel.

The merchant is considering changing the indexers mode to "Update on Schedule" from "Update on Save" to achieve better performance. However, the merchant is concerned about data consistency and the probability of the scheduled updates being lost.

How does Magento store the IDs of updated products for reindex in "Update on Schedule" mode?

## Options:

**A-** Magento is gathering updated IDs using catalog_product_save_commit_after events and publishes them into the Message Queue.

**B-** Magento sets a reindex_required flag on updated products and a later reindex run by the cron will pick up IDs based on that flag.

**C-** Using the Staging update scheduled for every minute, which includes modified products, Magento will automatically reindex affected products when the update is applied.

**D-** Magento enables database triggers which collect updated IDs and saves them in special changelog tables.

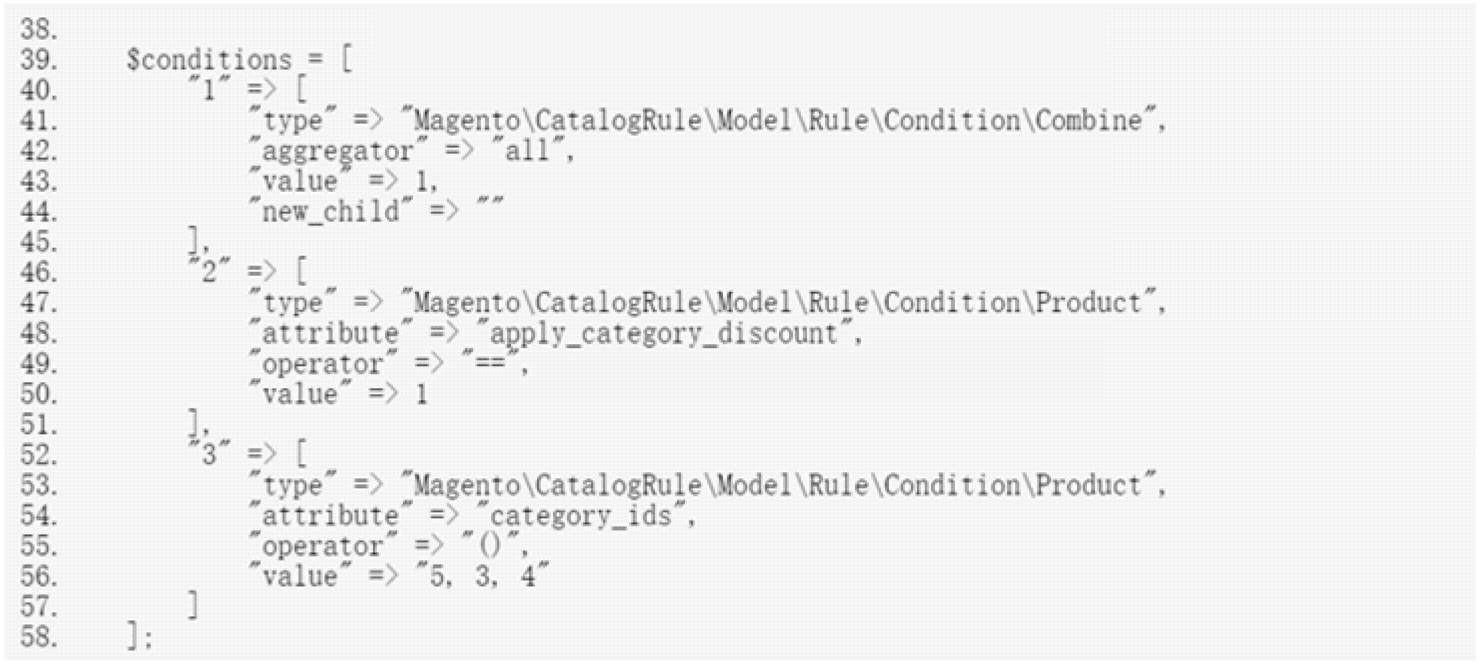
## Answer:

C

## Question 3

**Question Type:** MultipleChoice

You are creating a module that creates a Catalog Price Rule. You have written this code to specify the conditions:



```
    $conditions = [
        "1" => [
            "type" => "Magento\CatalogRule\Model\Rule\Condition\Combine",
            "aggregator" => "all",
            "value" => 1,
            "new_child" => ""
        ],
        "2" => [
            "type" => "Magento\CatalogRule\Model\Rule\Condition\Product",
            "attribute" => "apply_category_discount",
            "operator" => "==",
            "value" => 1
        ],
        "3" => [
            "type" => "Magento\CatalogRule\Model\Rule\Condition\Product",
            "attribute" => "category_ids",
            "operator" => "()",
            "value" => "5, 3, 4"
        ]
    ];
```

After saving the catalog rule, the second and third condition do not seem to fall under the combine condition.

How do you fix the issue?

## Options:

**A-** The type of the third condition needs to be Magento\CatalogRule\Model\Rule\Condition\Category since you are checking on category_ids

**B-** The array keys are incorrect they need to be 1--1 and 1--2

**C-** The value field in the first condition is changed to '2. 3'

**D-** The () operator in the third condition is invalid, which prevents the conditions from being saved correctly

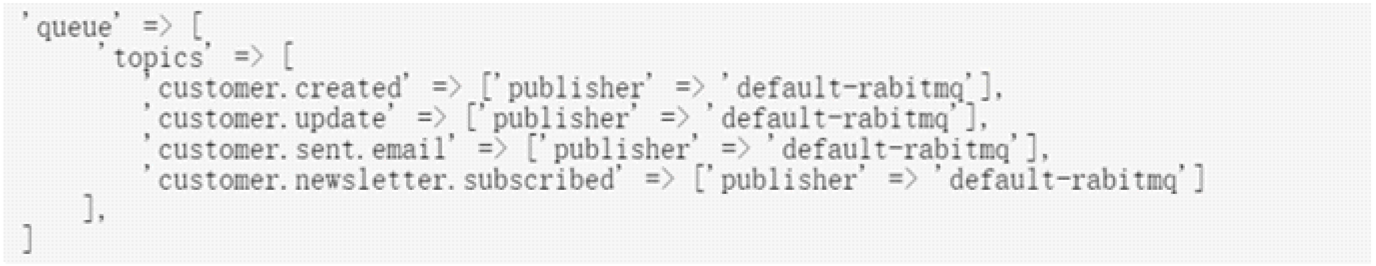
**Answer:**

D

## Question 4

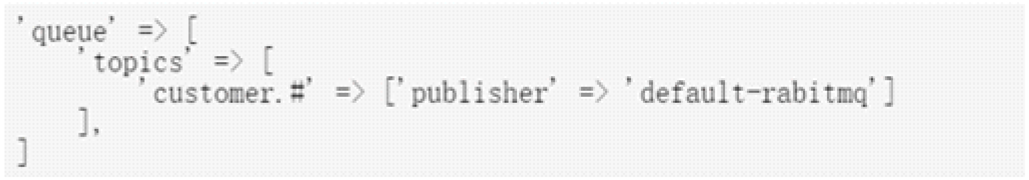
**Question Type:** MultipleChoice

You are tasked to work on a message queue module. When looking at the queue configuration you see the following:



```
'queue' => [
    'topics' => [
        'customer.created' => ['publisher' => 'default-rabitmq'],
        'customer.update' => ['publisher' => 'default-rabitmq'],
        'customer.sent.email' => ['publisher' => 'default-rabitmq'],
        'customer.newsletter.subscribed' => ['publisher' => 'default-rabitmq']
    ],
]
```

Keeping in mind maintainability, you change it to:



```
'queue' => [
    'topics' => [
        'customer.#' => ['publisher' => 'default-rabitmq']
    ],
]
```

What will happen?

### Options:

**A-** This code will cancel all the topic starting with customer

**B-** This code will not work as message queue does not allow # wildcard

**C-** This code will replace all of the different topics as # is the symbol for everything

**D-** This code will replace only some of the topics

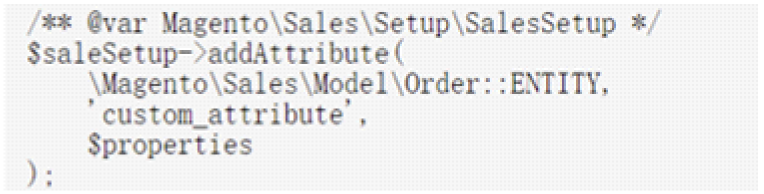
### Answer:

D

## Question 5

**Question Type:** MultipleChoice

While reviewing a module you found an upgrade script with the code:



```
/** @var Magento\Sales\Setup\SalesSetup */
$saleSetup->addAttribute(
    \Magento\Sales\Model\Order::ENTITY,
    'custom_attribute',
    $properties
);
```

What two actions will be done by the SalesSetup: :addAttribute method?

## Options:

**A-** The custom_attribute added to the eav_attribute table

**B-** The custom_attribute IS added to the order_eav_attribute table

**C-** The sales_order_grid table is altered with a new field custom_attribute

**D-** The sales_ordor table is altered with a new field custom_attribute

## Answer:

B, D

## Question 6

**Question Type:** **MultipleChoice**

You have a task to show a new EAV attribute of the customer entity on the customer_account_create form. Assume that the attribute already exists and its name is custom_attribute.

How do you add it to the form using a Data Patch script?

## Options:

**A-** This requires UI components customizations and can not be done with Data Patch

B)

```
/** @var Magento\Customer\Model\Customer\Form $form */
$form->addAttribute('customer_account_create', 'custom_attribute')
   ->save();
```

C)

```
/** @var Magento\Eav\Model\Config $eavConfig */
$eavConfig->getAttribute('customer', 'custom_attribute')
    ->setUsedInForms(['customer_account_create'])
    ->save();
```

D)

```
/** @var Magento\Customer\Api\AccountManagementInterface $accountManagement */
$accountManagement->addAttributeToForm('customer_account_create', 'custom_attribute');
```

**A-** Option A

**B-** Option B

**C-** Option C

**D-** Option D

### Answer:

C

## Question 7

**Question Type:** MultipleChoice

To prevent Cross Site Scripting (XSS) attacks, Magento templates use different methods to escape the output on the website before displaying it to the user. What three methods does Magento use to prevent this kind of attack?

### Options:

**A-** Sblock->escapeData()

**B-** Sblock->escapeOutput()

**C-** Sblock->escapeHtmlAttr()

**D-** $block->escapeHtml()

**E-** $block->escapeUrl()

**Answer:**

B, C, D

## Question 8

**Question Type:** **MultipleChoice**

On a merchant website you inherit an extension that allows customers to rate products. The extension adds two integer attributes to store the rating values, avg_rating and rating_count. The merchant reports an issue that the displayed average is lower than it should be. You determine you need to change the avg_rating attribute backend typetodecim to fix the issue.

In your module's Data Patch script, which three steps do you take to change the attribute backend type?

A)

Copy the `avg_rating values` from `catalog_product_entity_int` to `catalog_product_entity_decimal` with SQL

B)

```
$eavSetup->updateAttribute('product', 'avg_rating', 'backend_type', 'decimal');
```

C)

Delete the `avg_rating` values from `catalog_product_entity_int` with SQL

D)

```
$setup->getConnection()->migrate('catalog_product_entity_int', 'catalog_product_entity_decimal')
```

E)

```
$eavSetup->removeAttribute('product', 'avg_rating');
$eavSetup->addAttribute('product', 'avg_rating', $optionsWithDecimalBackendType);
```

## Options:

**A-** Option A

**B-** Option B

**C-** Option C

**D-** Option D

**E-** Option E

## Answer:

B, C, E