PROD

INTERNATIONAL SOFTWARE ENGINEERING OLYMPIAD

Version 2

Question 1 (2 points)

Select the correct statements. You can select multiple answers.

- Deserialization is the process of converting a data structure in a programming language into bytes
- Deserialization is the process of converting bytes into a data structure in a programming language
- Deserialization can be applied before receiving data over a network
- Deserialization is the process of sequential object creation

Question 2 (2 points)

Select the correct commands. You can select multiple answers.

- git --help
- git commit get ‘Add file’
- git remove file.txt
- git reset --hard
- git pull

Question 3 (1 points)

Select the correct statement.

- Method overriding is a mechanism that allows the use of methods with the same name but different parameters
- Method overriding is a mechanism that allows changing the behavior of a method in a child class
- Method overriding allows creating two identical methods with the same parameters and types, but with different return types

Question 4 (2 points)

Select the correct statements. You can select multiple answers.

- Cookies are passed in the request URL
- Cookies are passed in the HTTP request body
- Tokens and passwords can be stored in cookies
- Cookies are stored in the browser

Question 5 (2 points)

select the correct statements. You can select multiple answers.

- A graph is a tree
- In a graph, there are always more vertices than edges
- Breadth-first search can be used to find the distance between two vertices in a connected graph
- Depth-first search can be used to find the distance between two vertices in a connected graph

Question 6 (2 points)

Which of the following are valid CSS properties? You can select multiple answers.

- font-opacity
- image-opacity
- padding-left
- text-decoration
- background-position

Question 7 (2 points)

Select the correct statements. You can select multiple answers.

- A deque can be used as a stack
- Traversing a binary tree takes O(log N) time
- A heap data structure can be used to sort the elements stored in it
- Deleting from the middle of an array takes O(1) time

Question 8 (2 points)

Select the correct statements. You can select multiple answers.

- If the result of a hash function is known, the original hashed value can easily be retrieved
- A hash function is needed to search for elements in a binary search tree
- If the hash function results for two values are equal, the original values being hashed may still be different
- A hash function always returns a string value

Question 9 (2 points)

Which of the following can be used to store and perform operations on numbers of arbitrary size? You can select multiple answers.

- Arbitrary-precision arithmetic
- Rounding
- Two's complement representation
- Long, Int64, or Double data types

Question 10 (2 points)

Select the correct statements. You can select multiple answers.

- Python does not support object-oriented programming
- Java is an object-oriented programming language
- CSS is a functional programming language
- JavaScript can be used to write browser interfaces

QUESTION 11 (2 POINTS)

For which scenarios, according to REST, is it correct to use the POST method?

- Deleting a post
- Creating a new user
- Commenting on a post
- Updating an email address in a profile
- Retrieving a list of user messages

Question 12 (2 points)

Which of the following sorting algorithms are guaranteed to run in O(N log N)? You can select multiple answers.

- Heap sort
- Merge sort
- Quick sort
- Insertion sort

Question 13 (2 points)

Select the correct statements. You can select multiple answers.

When a browser and a backend server interact over HTTP, the following happens:

- The browser sends a request and is not required to wait for a response from the server
- The server can send requests to the browser on its own
- The browser is the first to send a request to the server and must wait for a response from it
- The browser can send multiple requests to the server simultaneously

Question 14 (2 points)

Select the UI container objects used for layout and positioning of other UI elements in mobile development. You can select multiple answers.

- VerticalStack
- Table/Grid Layout
- OrderedViews
- HorizontalQueue

Question 15 (2 points)

Select the correct statements. You can select multiple answers.

- Retrieving data from a database is faster than retrieving it from application memory
- Storing data in application memory is more reliable than storing it in a database
- If data is stored in a database and the web server stops working, the data will be preserved
- A database can be located on a separate server, independently from the backend application

Question 16 (2 points)

In which part of an HTTP request is authentication data passed? You can select multiple answers.

- Request header
- Request body
- Query parameters
- In the method name

Question 17 (1 points)

What is a media query?

- A JavaScript function for filtering multimedia files before they are loaded
- A resource storing multimedia files, accessible from a web browser
- A marker used to display the loading progress of multimedia content on a web page
- A method used to define and apply styles based on screen resolution and other device characteristics

Question 18 (2 points)

What is the purpose of signing a mobile application? You can select multiple answers.

- To confirm authorship and authenticity
- For security purposes
- To add dynamic libraries
- To optimize the application

Question 19 (2 points)

Select the correct statements related to horizontal scaling. You can select multiple answers.

- Increasing the amount of RAM on a server
- Increasing the number of CPU cores
- Increasing the number of servers
- Increasing the number of database instances

Question 20 (2 points)

Which of the following pseudo-elements exist in CSS? You can select multiple answers.

- visited
- after
- first-letter
- read-write

Question 21 (2 points)

Select the correct statements. You can select multiple answers.

- Retrieving data from a hash table runs in O(1)
- Saving data to a hash table runs in O(log N)
- Traversing all elements of a binary tree runs in O(log N)
- Retrieving the smallest element from a heap runs in O(1)

Question 22 (2 points)

Which data structures exist in JavaScript according to the current standard?
You can select multiple answers.

- Queue
- Proxy
- Tuple
- WeakSet
- Array

Question 23 (2 points)

Select the correct statements. You can select multiple answers.

- A backend application can handle multiple user requests simultaneously
- A mobile application cannot function without a server side
- A backend application can operate without a database
- A frontend application can cache data in the browser

Question 24 (2 points)

Which APIs does the browser provide? You can select multiple answers.

- Concurrency API
- Mouse API
- Canvas API
- Keyboard API
- Page Visibility API

Question 25 (5 points)

The save_all_values function takes a long time to execute because it saves data to remote storage, so we want to call it only when 20 elements have accumulated in the value list. Select the correct implementation for saving values using buffering.

- def bufferize _ save(value):
  values.append(value)
  if len(values) == 20:
  save _ all _ values(values)
- def bufferize _ save(value):
  values.append(value)
  if len(values) == 20:
  save _ all _ values(values)
  values.clear()
- def bufferize _ save(value):
  values.append(value)
  if len(values) == 20:
  save _ all _ values(values)
  values.clear()
- def bufferize _ save(value):
  values.append(value)
  if len(values) < 20:
  save _ all _ values(values)
  values.clear

Question 26 (5 points)

Imagine you want to implement a service for analyzing product price data. To do this, call an external product API over HTTP, save the data to a database, and then display it to the client. Select the correct code version where all possible errors are handled properly:

- get _ price(product):
  price = None
  try:
  price = http _ client.get('https://prodcontest.productprices.ru/' + product)
  except:
  return 'Error when retrieving the product price from the service' database.save _ price(product, price)
  return 'Product price: ' + price

- def get _ price(product):
  price = None
  try:
  price = http _ client.get('https://prodcontest.productprices.ru/' + product)
  except:
  return 'Error when retrieving the product price from the pricing service'
  try:
  database.save _ price(product, price)
  except:
  return 'Error when saving the product price to the database'
  return 'Product price: ' + price

- def get _ price(product):
  price = http _ client.get('https://prodcontest.product-prices.ru/' + product)
  try:
  database.save _ price(product, price)
  except:
  return 'Error when saving the product price to the database' return 'Product price: ' + price

- def get _ price(product):
  price = None
  try:
  price = http _ client.get('https://prodcontest.productprices.ru/' + product)
  database.save _ price(product, price)
  except:
  return 'Product price: ' + price
  return 'Product price: ' + price

Question 27 (6 points)

There is an application for booking theater tickets. To make the service convenient, push notifications about changes in order status are sent to users. Select the correct code options that prevent sending duplicate notifications if the order status remains unchanged

- def update _ order _ status(order _ id, status):
  order = database.get _ order(order _ id)
  order.status = status
  database.save(order)
  send(order.client _ id, 'Order status changed to ' + status)
- def update _ order _ status(order _ id, status):
  order = database.get _ order(order _ id)
  current _ status = order.status
  order.status = status
  database.save(order)
  if current _ status == status:
  send(order.client _ id, 'Order status changed to ' + status)
- def update _ order _ status(order _ id, status):
  order = database.get _ order(order _ id)
  current _ status = order.status
  order.status = status database.save(order)
  if current _ status != status:
  send(order.client _ id, 'Order status changed to ' + status)
- def update _ order _ status(order _ id, status):
  order = database.get _ order(order _ id)
  order.status = status
  database.save(order)
  if order.status != status:
  send(order.client _ id, 'Order status changed to ' + status)

Question 28 (6 points)

After the introduction of a new cashback service, it was discovered that some features work slowly: users were leaving because certain screens took too long to load.

The developers found that the problem was caused by a REST method that returns the amount of cashback earned by the user. To fix the issue, they decided to use data caching to avoid querying the database on every request. Select the correct implementations for retrieving and updating cashback data using caching.

- def get _ cashback(client _ id):
  if not cache.exists(client _ id):
  cashback _ amount = database.get _ cashback(client _ id)
  cache.put(client _ id, cashback _ amount)
  return cache.get(client _ id)
  def update _ cashback(client _ id, cashback _ amount):
  database.save _ cashback(client _ id, cashback _ amount)
  cache.put(client _ id, cashback _ amount)

- ```
  def get _ cashback(client _ id):
      if cache.exists(client _ id):
          cashback _ amount = database.get _ cashback(client _ id)
          cache.put(client _ id, cashback _ amount)
      return cache.get(client _ id)
  def update _ cashback(client _ id, cashback _ amount):
      database.save _ cashback(client _ id, cashback _ amount)
      cache.put(client _ id, cashback _ amount)
  ```

- ```
  def get _ cashback(client _ id):
      if cache.exists(client _ id):
          return cache.get(client _ id)
      cashback _ amount = database.get _ cashback(client _ id)
      cache.put(client _ id, cashback _ amount)
      return cashback _ amount
  def update _ cashback(client _ id, cashback _ amount):
      database.save _ cashback(client _ id, cashback _ amount)
      cache.put(client _ id, cashback _ amount)
  ```

- ```
  def get _ cashback(client _ id):
      if not cache.exists(client _ id):
          cashback _ amount = database.get _ cashback(client _ id)
      cache.put(client _ id, cashback _ amount)
          return cache.get(client _ id)
  def update _ cashback(client _ id, cashback _ amount):
      cache.put(client _ id, cashback_amount)
  ```

Question 29 (16 points)

Imagine you're writing a module for managing a user's shopping cart in an online grocery store where the quantity of each item is limited. You need to implement a program that adds and removes products from the user's cart.

Input format

The first line contains an integer N (1 < N < 105), representing the total number of items in the store. The following N lines contain pairs of values ai and bi, where ai is the product name (length not exceeding 105 characters; product names are always unique strings) and bi is the quantity of that product (1 < bi < 103). Next comes an integer T (1 < T < 105), representing the number of user actions. The following T lines contain xi–yi pairs, where xi takes one of two values: "+" to add a product or "-" to remove a product, and yi is the product name. If a product being added to the cart is no longer available in the store, it's not added. Remove operations will only be given for products that are already in the cart.

Output format

For each line of output, print an ai–bi pair, where ai is the product name (always a unique string) and bi is the quantity of that product in the customer's cart. Print only those products that are currently in the cart, sorted in alphabetical order.

Example 1

Input, Output
  ```