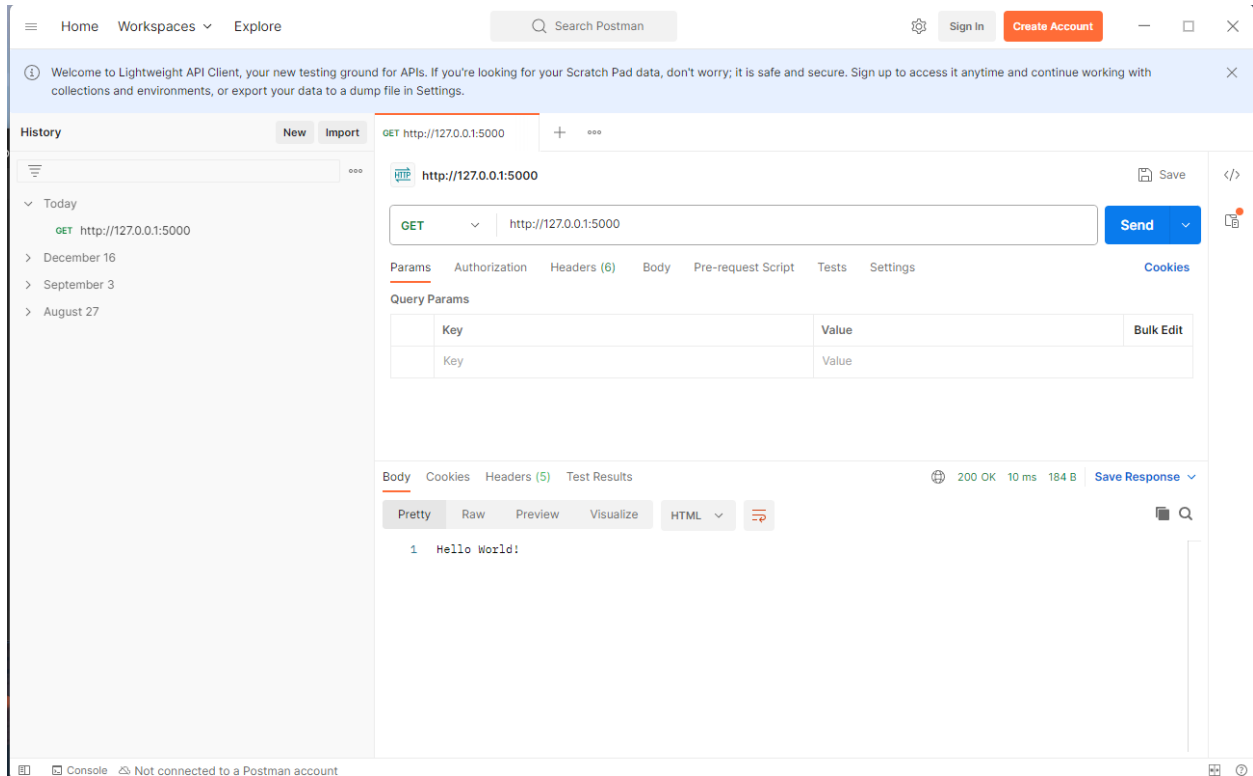


BizKit Interview Task Test Samples

Local environment setup:

```
PS D:\Lex\programming\python\lextermalit-phasebook> Set-ExecutionPolicy Unrestricted -Scope Process
PS D:\Lex\programming\python\lextermalit-phasebook> py -3 -m venv venv
PS D:\Lex\programming\python\lextermalit-phasebook> venv\Scripts\activate
(venv) PS D:\Lex\programming\python\lextermalit-phasebook> pip install -r requirements.txt
```



Match Algorithm Optimization:

Before Optimization:

Match Case 0

The screenshot displays the Postman API client interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' tabs, along with a search bar and 'Sign In'/'Create Account' buttons. A welcome message for 'Lightweight API Client' is shown below the navigation bar.

The main interface is divided into three sections:

- History:** A list of recent requests on the left, including 'GET http://127.0.0.1:5000/match/0'.
- Request Editor:** The central area where the current request is configured. It shows a 'GET' method and the URL 'http://127.0.0.1:5000/match/0'. Below this, there are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Query Params' section is currently active, showing a table with 'Key' and 'Value' columns.
- Response Viewer:** The bottom section displays the response from the server. It shows a status of '200 OK', a response time of '7 ms', and a body size of '217 B'. The response body is formatted as JSON, showing an object with 'elapsedTime' and 'message' properties.

The response body is displayed in a 'Pretty' format, showing the following JSON structure:

```
1 {
2   "elapsedTime": 0.0,
3   "message": "Match found"
4 }
```

The status bar at the bottom indicates 'Console' and 'Not connected to a Postman account'.

Match Case 1

The screenshot displays the Postman API client interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options, along with a search bar and 'Sign In'/'Create Account' buttons. A welcome message for 'Lightweight API Client' is shown below the navigation bar.

The main interface is divided into three sections:

- History:** A sidebar on the left showing a list of recent requests. Under 'Today', there are four GET requests to `http://127.0.0.1:5000/match/1`, `http://127.0.0.1:5000/match/0`, `http://127.0.0.1:5000/match/3`, and `http://127.0.0.1:5000`. Below this, dates 'December 16', 'September 3', and 'August 27' are listed.
- Request Editor:** The central area shows a GET request to `http://127.0.0.1:5000/match/1`. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Query Params' section is currently active, showing a table with two columns: 'Key' and 'Value'. The first row has 'Key' in the 'Key' column and 'Value' in the 'Value' column. A 'Bulk Edit' button is visible on the right.
- Response View:** The bottom section shows the response details. It includes tabs for 'Body', 'Cookies', 'Headers (5)', and 'Test Results'. The 'Body' tab is selected, showing a JSON response in 'Pretty' format. The response is:

```
{  "elapsedTime": 0.0,  "message": "No match"}
```

. The status bar at the bottom right indicates '200 OK', '11 ms', and '214 B'. A 'Save Response' button is also present.

At the very bottom, there's a status bar with 'Console' and 'Not connected to a Postman account' indicators.

Match Case 2

Home Workspaces Explore

ⓘ

Welcome to Lightweight API Client, your new testing ground for APIs. If you're looking for your Scratch Pad data, don't worry; it is safe and secure. Sign up to access it anytime and continue working with collections and environments, or export your data to a dump file in Settings.

×

History

New Import

GET http://127.0.0.1:5000

GET http://127.0.0.1:5000/match/2

+

...

⌵

...

Today

GET http://127.0.0.1:5000/match/2

GET http://127.0.0.1:5000/match/1

GET http://127.0.0.1:5000/match/0

GET http://127.0.0.1:5000/match/3

GET http://127.0.0.1:5000

> December 16

> September 3

> August 27

http://127.0.0.1:5000/match/2

Save

</>

GET

http://127.0.0.1:5000/match/2

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

Cookies

Headers (5)

Test Results

200 OK 56.04 s 231 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

⌵

⌵

1

2

3

4

⌵

"elapsedTime": 56.02891540527344,

"message": "Match found"

Console

Not connected to a Postman account

⌵ ⓘ

Match Case 3

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' tabs, a search bar, and 'Sign In' and 'Create Account' buttons. A welcome message is displayed below the navigation bar.

The main interface is divided into three sections:

- History:** A list of recent requests on the left side, including several GET requests to `http://127.0.0.1:5000/match/3`, `http://127.0.0.1:5000/match/2`, `http://127.0.0.1:5000/match/1`, `http://127.0.0.1:5000/match/0`, and `http://127.0.0.1:5000`.
- Request Editor:** The central area where a new request is being configured. The URL is `http://127.0.0.1:5000/match/2` and the method is `GET`. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Params' tab is active, showing a table for 'Query Params' with columns 'Key' and 'Value'. A 'Bulk Edit' button is also present.
- Response View:** The bottom section displays the response from the server. It includes tabs for 'Body', 'Cookies', 'Headers (5)', and 'Test Results'. The 'Body' tab is active, showing a JSON response in 'Pretty' format. The response is:

```
{  "elapsedTime": 52.00103807449341,  "message": "No match"}
```

. The status bar at the bottom indicates a 200 OK response with a response time of 52.01 s and a size of 228 B.

At the bottom of the interface, there is a status bar showing 'Console' and 'Not connected to a Postman account'.

After Optimization:

Match Case 0:

The screenshot displays the Postman API client interface. The top navigation bar includes links for Home, Workspaces, and Explore, along with a search bar and buttons for Sign In and Create Account. A welcome message is shown below the navigation bar.

The main interface is divided into three sections:

- History:** A list of recent requests, including several GET requests to `http://127.0.0.1:5000/match/0` and `http://127.0.0.1:5000/match/2`.
- Request Editor:** The active request is a GET request to `http://127.0.0.1:5000/match/2`. The URL bar shows the method (GET) and the full URL. Below the URL bar, there are tabs for Params, Authorization, Headers (6), Body, Pre-request Script, Tests, and Settings. The Params tab is currently selected, showing a table with one row: Key (Key) and Value (Value).
- Response Viewer:** The response is displayed in the bottom section. It shows a status of 200 OK, a response time of 7 ms, and a body size of 217 B. The response body is shown in JSON format, containing the following data:

```
{  "elapsedTime": 0.0,  "message": "Match found"}
```

The bottom status bar indicates that the client is not connected to a Postman account.

Match Case 1:

The screenshot displays the Postman API client interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' tabs, along with a search bar and user options. A welcome message is shown below the navigation bar. The main interface is divided into three sections: a left sidebar for history, a central workspace for editing requests, and a right sidebar for response details.

History Sidebar: Lists recent requests under 'Today' and previous dates. The first entry is a GET request to `http://127.0.0.1:5000/match/1`.

Central Workspace: Shows the selected request, which is a GET request to `http://127.0.0.1:5000/match/2`. The request is configured with the following details:

- Method:** GET
- URL:** `http://127.0.0.1:5000/match/1`
- Params:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings
- Query Params:** A table with two columns: Key and Value. The first row has 'Key' in the Key column and 'Value' in the Value column.

Response Details: The response is shown in the right sidebar. It includes the status code '200 OK', the time taken '7 ms', and the size '214 B'. The response body is displayed in 'Pretty' format, showing a JSON object:

```
1 {
2   "elapsedTime": 0.0,
3   "message": "No match"
4 }
```

The bottom status bar indicates 'Console' and 'Not connected to a Postman account'.

Match Case 2:

The screenshot displays the Postman API client interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options, along with a search bar and user account links. A welcome message is shown below the navigation bar. The main interface is divided into three sections: History, Request Editor, and Response Viewer.

History: A list of recent requests is shown, including several GET requests to `http://127.0.0.1:5000/match/2` and `http://127.0.0.1:5000/match/0`.

Request Editor: The selected request is a GET request to `http://127.0.0.1:5000/match/2`. The 'Query Params' section is visible, showing a table with one entry: 'Key' and 'Value'.

Key	Value
Key	Value

Response Viewer: The response is displayed in 'Pretty' format, showing a JSON object with 'elapsedTime' and 'message' fields.

```
1 {
2   "elapsedTime": 0.01364183425983283,
3   "message": "Match found"
4 }
```

The status bar at the bottom indicates '200 OK', '23 ms', and '234 B'.

Match Case 3:

Home Workspaces Explore Search Postman Sign In Create Account

Welcome to Lightweight API Client, your new testing ground for APIs. If you're looking for your Scratch Pad data, don't worry; it is safe and secure. Sign up to access it anytime and continue working with collections and environments, or export your data to a dump file in Settings.

History New Import GET http://127.0.0.1:5000 GET http://127.0.0.1:5000/match GET http://127.0.0.1:5000/match/2 + ...

Today

- GET http://127.0.0.1:5000/match/3
- GET http://127.0.0.1:5000/match/2
- GET http://127.0.0.1:5000/match/1
- GET http://127.0.0.1:5000/match/0
- GET http://127.0.0.1:5000/match/3
- GET http://127.0.0.1:5000/match/2
- GET http://127.0.0.1:5000/match/1
- GET http://127.0.0.1:5000/match/0
- GET http://127.0.0.1:5000/match/3
- GET http://127.0.0.1:5000

> December 16

> September 3

> August 27

http://127.0.0.1:5000/match/2

GET http://127.0.0.1:5000/match/3 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (5) Test Results 200 OK 23 ms 231 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "elapsedTime": 0.017043352127075195,
3   "message": "No match"
4 }
```

Console Not connected to a Postman account

Implement Search Algorithm:

No Parameter Request: http://127.0.0.1:5000/search?

http://127.0.0.1:5000/search?id=1 Save

GET http://127.0.0.1:5000/search? Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params



Key	Value	Bulk Edit
<input checked="" type="checkbox"/>		
Key	Value	

Body Cookies Headers (5) Test Results 200 OK 9 ms 725 B Save Response

Pretty Raw Preview Visualize

```
[ { "age": 29, "id": "1", "name": "John Doe", "occupation": "Developer" }, { "age": 30, "id": "2", "name": "Jane Doe", "occupation": "Engineer" }, { "age": 25, "id": "3", "name": "Joe Doe", "occupation": "Designer" }, { "age": 28, "id": "4", "name": "John Smith", "occupation": "Architect" }, { "age": 31, "id": "5", "name": "Jane Smith", "occupation": "Manager" }, { "age": 24, "id": "6", "name": "Joe Smith", "occupation": "Designer" } ]
```

- **Request:** http://127.0.0.1:5000/search?id=1

 http://127.0.0.1:5000/search?id=1  Save

GET


http://127.0.0.1:5000/search?id=1


Send

Params • Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	id	1	
	Key	Value	

Body Cookies Headers (5) Test Results  200 OK 6 ms 260 B [Save Response](#)

Pretty Raw Preview Visualize JSON 

```
1 {
2   {
3     "age": 29,
4     "id": "1",
5     "name": "John Doe",
6     "occupation": "Developer"
7   }
8 }
```

- **Request:** http://127.0.0.1:5000/search?id=2&name=John

HTTP

http://127.0.0.1:5000/search?id=1

Save

GET

http://127.0.0.1:5000/search?id=2&name=John

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	id	2	
<input checked="" type="checkbox"/>	name	John	
	Key	Value	

Body

Cookies

Headers (5)

Test Results

200 OK

7 ms

448 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    {
3      "age": 29,
4      "id": "1",
5      "name": "John Doe",
6      "occupation": "Developer"
7    },
8    {
9      "age": 30,
10     "id": "2",
11     "name": "Jane Doe",
12     "occupation": "Engineer"
13   },
14   {
15     "age": 28,
16     "id": "3",
17     "name": "John Smith",
18     "occupation": "Architect"
19   }
20 ]
```

- **Request:** http://127.0.0.1:5000/search?age=28

HTTP

http://127.0.0.1:5000/search?id=1

Save

GET

http://127.0.0.1:5000/search?age=28

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	age	28	

Body

Cookies

Headers (5)

Test Results

200 OK6 ms356 B

Save Response

Pretty

Raw



Preview


Visualize

JSON


```
1  {
2    {
3      "age": 29,
4      "id": "1",
5      "name": "John Doe",
6      "occupation": "Developer"
7    },
8    {
9      "age": 28,
10     "id": "4",
11     "name": "John Smith",
12     "occupation": "Architect"
13   }
14 }
```


- **Request:** http://127.0.0.1:5000/search?id=5&name=Joe&age=30&occupation=Arc

 http://127.0.0.1:5000/search?id=1  Save



GET 

http://127.0.0.1:5000/search?id=5&name=Joe&age=30&occupation=Arc

Send 

Params  Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

<input checked="" type="checkbox"/>	age	30
<input checked="" type="checkbox"/>	occupation	Arc
	Key	Value

Body Cookies Headers (5) Test Results  200 OK 8 ms 725 B [Save Response](#) 

Pretty Raw Preview Visualize

[{ "age": 29, "id": "1", "name": "John Doe", "occupation": "Developer" }, { "age": 30, "id": "2", "name": "Jane Doe", "occupation": "Engineer" }, { "age": 25, "id": "3", "name": "Joe Doe", "occupation": "Designer" }, { "age": 28, "id": "4", "name": "John Smith", "occupation": "Architect" }, { "age": 31, "id": "5", "name": "Jane Smith", "occupation": "Manager" }, { "age": 24, "id": "6", "name": "Joe Smith", "occupation": "Designer" }]

Bonus Challenge:

Request: `http://127.0.0.1:5000/search?id=5&name=Joe&age=30&occupation=Arc`

HTTP

http://127.0.0.1:5000/search?id=1

Save

GET

http://127.0.0.1:5000/search?id=5&name=Joe&age=30&occupation=Arc

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	id	5	
<input checked="" type="checkbox"/>	name	Joe	
<input checked="" type="checkbox"/>	age	30	
<input checked="" type="checkbox"/>	occupation	Arc	
	Key	Value	

Body

Cookies

Headers (5)

Test Results

200 OK6 ms725 BSave Response

Pretty

Raw

Preview

Visualize

```
[ { "age": 31, "id": "5", "name": "Jane Smith", "occupation": "Manager" }, { "age": 25, "id": "3", "name": "Joe Doe", "occupation": "Designer" }, { "age": 24, "id": "6", "name": "Joe Smith", "occupation": "Designer" }, { "age": 29, "id": "1", "name": "John Doe", "occupation": "Developer" }, { "age": 30, "id": "2", "name": "Jane Doe", "occupation": "Engineer" }, { "age": 28, "id": "4", "name": "John Smith", "occupation": "Architect" } ]
```