from flask import Flask, request, jsonify

app = Flask(\_\_name\_\_)

@app.route('/bfhl', methods=['POST'])

def process\_data():

"""

Processes an array of data to identify and categorize numbers, alphabets,

and special characters.

Method: POST

Route: /bfhl

"""

try:

# Get data from the request body

data = request.get\_json().get('data')

# Basic validation

if not isinstance(data, list):

return jsonify({

"is\_success": False,

"user\_id": "aryan\_garg\_29072025",

"error": "Input data must be an array."

}), 400

# Initialize response lists and variables

odd\_numbers = []

even\_numbers = []

alphabets = []

special\_characters = []

all\_alpha\_chars = []

total\_sum = 0

# Process each item in the input array

for item in data:

if isinstance(item, str):

# Check if item is a number

if item.isnumeric():

num = int(item)

total\_sum += num

if num % 2 == 0:

even\_numbers.append(str(num)) # Numbers must be returned as strings

else:

odd\_numbers.append(str(num)) # Numbers must be returned as strings

# Check if item is an alphabet or a word

elif item.isalpha():

alphabets.append(item.upper()) # Convert alphabets to uppercase

all\_alpha\_chars.extend(list(item))

# Otherwise, it's a special character

else:

special\_characters.append(item)

else:

# Handle non-string elements if necessary, here we consider them special characters

special\_characters.append(str(item))

# Generate the concatenated string with alternating caps

all\_alpha\_chars.reverse()

concat\_list = []

for i, char in enumerate(all\_alpha\_chars):

if i % 2 == 0:

concat\_list.append(char.upper())

else:

concat\_list.append(char.lower())

concat\_string = "".join(concat\_list)

# Construct the successful response object

response = {

"is\_success": True, [cite: 5]

"user\_id": "aryan\_garg\_29072025", [cite: 4]

"email": "aryan.garg.official@chitkara.edu.in", [cite: 8]

"roll\_number": "2110991901", [cite: 8]

"odd\_numbers": odd\_numbers, [cite: 1]

"even\_numbers": even\_numbers, [cite: 1]

"alphabets": alphabets, [cite: 1]

"special\_characters": special\_characters, [cite: 1]

"sum": str(total\_sum), # Return sum as a string

"concat\_string": concat\_string [cite: 1]

}

# Return the JSON response with a 200 status code for success

return jsonify(response), 200

except Exception as e:

# Graceful exception handling

return jsonify({

"is\_success": False,

"user\_id": "aryan\_garg\_29072025",

"error": f"An error occurred: {str(e)}"

}), 500

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True, port=5000)