1.Reverse a Stack You are given a stack of integers. Your task is to reverse the order of the elements in the stack using only stack operations (push and pop) and without using any additional data structures. Ex. stack = [1, 2, 3, 4, reverse Stack(stack) print(stack) Output should be [5, 4, 3, 2, 1].

Ans: qThis way, the first element becomes the last, and the last becomes the first

Code:

def insert\_at\_bottom(stack, item):

if not stack:

stack.append(item)

else:

top = stack.pop()

insert\_at\_bottom(stack, item)

stack.append(top)

def reverse\_stack(stack):

if stack:

top = stack.pop()

reverse\_stack(stack)

insert\_at\_bottom(stack, top)

# Example

stack = [1, 2, 3, 4, 5]

reverse\_stack(stack)

print(stack)

Output will be:

[5, 4, 3, 2, 1]

2.Three Sum Problem Sample Problem: Given an array of Integers, find all unique triplets in the array which give the sum of Zero. The solution should return the list of triplets.

Ans:

1. Sort the array to make it easier to find triplets and avoid duplicates.

2. Loop through the array, choosing each number as the first number of the triplet.

3. For each number, use two pointers: one at the next number, and one at the end.

4. Check if the sum of the three numbers is zero.

5. If the sum is zero, add the triplet to the result.

6. Move the pointers inward to find new combinations.

7. Skip duplicate numbers to avoid repeating the same triplet.

8. If the sum is less than zero, move the left pointer to increase the sum.

9. If the sum is more than zero, move the right pointer to decrease the sum.

10. Return the list of all unique triplets.

3.Depth-First Search (DFS) Sample Problem: Implement Depth-First Search (DFS) to traverse a graph starting from a given vertex. The graph is represented by an adjacency list.

Ans:1. DFS starts at a given node (or vertex).

2. Use an adjacency list to represent the graph.

3. Use a recursive function to explore the graph.

4. Mark the starting node as visited so you don’t revisit it.

5. Visit all unvisited neighbors of the current node.

6. For each unvisited neighbor, call DFS recursively to continue exploring.

7. Continue until all nodes connected to the starting node are visited.

8. The traversal order is the result of the DFS.

4.Create a Simple Website with the Following Features: a. Display A welcoming message and a brief description. b. Include navigation links to the homepage, about page, contact page, and blog page.

Ans:

<!DOCTYPE html>

<html>

<head>

<title>My Simple Website</title>

</head>

<body>

<h1>Welcome to My Website!</h1>

<p>This is a brief description of my site.</p>

<nav>

<a href=”index.html”>Home</a> |

<a href=”about.html”>About</a> |

<a href=”contact.html”>Contact</a> |

<a href=”blog.html”>Blog</a>

</nav>

</body>

</html>

5.Implement a responsive social media platform using HTML, CSS, And JavaScript. Ensure that the UI displays properly on both Desktop and mobile devices, with features like swipe gestures for Navigation on touchenabled devices.

Ans:

<!DOCTYPE html>

<html>

<head>

<meta name=”viewport” content=”width=device-width, initial-scale=1.0”>

<style>

Body { font-family: Arial; margin: 0; }

.navbar { background-color: #333; padding: 1em; }

.navbar a { color: white; margin: 0 10px; text-decoration: none; }

.post { padding: 1em; border-bottom: 1px solid #ccc; }

@media (max-width: 600px) { .navbar { text-align: center; } } /\* Responsive \*/

</style>

</head>

<body>

<div class=”navbar”>

<a href=”#”>Home</a><a href=”#”>Profile</a><a href=”#”>Messages</a>

</div>

<div id=”posts” class=”post”>Swipe left or right to navigate</div>

<script>

Let posts = [‘Post 1’, ‘Post 2’, ‘Post 3’];

Let current = 0;

Document.body.addEventListener(‘touchstart’, e => start = e.touches[0].clientX);

Document.body.addEventListener(‘touchend’, e => {

If (e.changedTouches[0].clientX < start) current = (current + 1) % posts.length;

Else current = (current – 1 + posts.length) % posts.length;

Document.getElementById(‘posts’).innerText = posts[current];

});

</script>

</body>

</html>