COLLAGE OF BUSSINESS AND ECONOMICS

DEPARTIMENT OF BIT(BUSSIONESS INFORMATION TECHNOLOGY)

Name: BANKUNDIYE SYLVIE

REGNUMBER:224013635

DATA STRUCTURE AND ALGORITHM ASSIOGNIMENT

STACK

Q1   *MTN MoMo back button as LIFO*
     When you enter payment details step by step, each detail is pushed onto the stack. Pressing back   removes the last entered step first. This shows LIFO because the most recent step (last in) is the first removed (first out).

Q2.When you go back in Canvas, the most recent page visited is undone/removed first, just like popping the top element from a stack. The earlier steps remain underneath.

Q3.If each action (transaction, correction, typing,) is pushed to a stack, then undoing means

popping the last action. The stack ensures you can go backward step-by-step in the correct order.

Q4.Stacks ensure correctness by pushing each opening bracket (or field) and popping when a

matching closing bracket appears. If at the end the stack is empty, the form fields are balanced and properly matched.

Q5. *Task sequence*

Steps:

-Push("CBE notes") → stack = [CBE notes]

-Push("Math revision") → stack = [CBE notes, Math revision]

-Push("Debate") → stack = [CBE notes, Math revision, Debate]

-Pop() → removes "Debate" → stack = [CBE notes, Math revision]

-Push("Group assignment") → stack = [CBE notes, Math revision, Group assignment]

Top of stack = "Group assignment"

Q6.If the student pops (undoes) 3 actions, then the last 3 answers are removed, leaving only the earlier answers still in the stack. This ensures only the oldest unchanged work remains.

Q7. *Backtracking in RwandAir booking*

Each step of filling the form is pushed. To backtrack, you pop steps one by one, retracing exactly in reverse order. That's why stacks fit perfectly for step-by-step back navigation.

Q8. *Reversing proverb*

Phrase = "Umwana ni umutware"

Steps:

- Push "Umwana", Push "ni", Push "umutware"
- Pop → "umutware"
- Pop → "ni"
- Pop → "Umwana"

Reversed = "umutware ni Umwana"

Q9.DFS uses a stack because it goes deep into one shelf first, pushing unexplored paths, and only backtracks (popping) when no further depth is possible. A queue would explore breadth-first, which isn't efficient when searching deep.

Q10.A stack-based transaction navigation could allow users to press undo to backtrack through recent transactions or corrections, ensuring they can retrace steps safely before final confirmation.

QUEUE

Q1. *Restaurant serving = FIFO*
First customer in line is served first. This is First-In-First-Out — earliest arrival gets priority.

Q2. *YouTube autoplay as dequeue*
When a video finishes, the front of the playlist queue is dequeued. The next video (now at the front) plays automatically. This is the same as removing the first element in a q**Q3.** *RRA tax*
People arrive one after another and are served in their arrival order. This is a perfect example of enqueue and dequeue.

Q4. *Customer service improvement*
Queues ensure fairness — requests are processed in order, preventing confusion or jumping the line. This reduces frustration and keeps the system organized.

Q5. *Bank sequence*

-Enqueue("Alice"), Enqueue("Eric"), Enqueue("Chantal") → Queue = [Alice, Eric, Chantal]

-Dequeue() → removes Alice → Queue = [Eric, Chantal]

-Enqueue("Jean") → Queue = [Eric, Chantal, Front = Eric

Q6. *Fairness in pension applications*

Since a queue processes applications in arrival order, no one can skip ahead. Every applicant is handled fairly, ensuring orderliness.

Q7. *Queue types in Rwanda*

    -Linear queue = people waiting at a wedding buffet (first to line up eats first).

    -Circular queue = buses looping at Nyabugogo (after finishing, they rejoin at the back).

    -Deque = boarding a bus from either front or rear door (two-sided entry).

Q8. *Restaurant orders*

Orders are enqueued as customers place them. When food is ready, it is dequeued in the same order to call customers fairly.

Q9. *Priority queue at CHUK*

Emergency patients are prioritized ahead of normal arrivals. Unlike a normal queue (strict FIFO), priority queues reorder items by urgency.

Q10. *Moto/e-bike taxi app*

Drivers (supply) wait in a queue. Students (demand) are matched in order of arrival to the first available driver, ensuring fairness and avoiding bias.