

Samir Varma

3/7/2025

Programming Methodology 1

Professor Jorge Ortiz

Lab 7

Exercise 1

Basic Operations

a) Initialize database

```
● samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex1
All Students:
ID: 12345
Name: John Doe
Major: Computer Science
GPA: 3.75

ID: 67890
Name: Jane Smith
Major: Mathematics
GPA: 3.95

ID: 11223
Name: Alice Johnson
Major: Physics
GPA: 3.85
```

b) Search student by ID

```
Searching for ID 12345:
Found: John Doe
```

c) Remove student:

```
After Removing Student with ID 67890:
```

```
ID: 12345
```

```
Name: John Doe
```

```
Major: Computer Science
```

```
GPA: 3.75
```

```
ID: 11223
```

```
Name: Alice Johnson
```

```
Major: Physics
```

```
GPA: 3.85
```

Advanced Operations

a) GPA Thresholds

```
Students with GPA >= 3.8:
```

```
ID: 67890
```

```
Name: Jane Smith
```

```
Major: Mathematics
```

```
GPA: 3.95
```

```
ID: 11223
```

```
Name: Alice Johnson
```

```
Major: Physics
```

```
GPA: 3.85
```

b) Majors

```
Students with Major = Physics:  
ID: 11223  
Name: Alice Johnson  
Major: Physics  
GPA: 3.85
```

c) Sort by GPA (After removing John Doe)

```
Sorted Students by GPA:  
ID: 11223  
Name: Alice Johnson  
Major: Physics  
GPA: 3.85
```

```
ID: 12345  
Name: John Doe  
Major: Computer Science  
GPA: 3.75
```

Edge Cases

a) Empty Database (Full Database tested above, capacity = 3)

```
Students with GPA >= 3.8:
```

```
Students with Major = Physics:
```

```
Before Removing Student with ID 67890:
```

```
After Removing Student with ID 67890:
```

b) Searching for + Removing nonexistent student

```
Students with Major = English:
```

```
After Removing Student with ID 67891:
```

```
ID: 12345
```

```
Name: John Doe
```

```
Major: Computer Science
```

```
GPA: 3.75
```

```
ID: 67890
```

```
Name: Jane Smith
```

```
Major: Mathematics
```

```
GPA: 3.95
```

```
ID: 11223
```

```
Name: Alice Johnson
```

```
Major: Physics
```

```
GPA: 3.85
```

c) Invalid input (e.g. negative input)

```
⊗ samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex1
Invalid capacity.
All Students:
zsh: segmentation fault ./ex1
○ samirvarma@Samirs-MacBook-Pro-2 Lab7 %
```

Exercise 2

Basic Operations

a) Creating/Storing various arrays

```
• samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex2
Dynamic Array:
Data: [10]
Size: 1
Capacity: 2
```

```
• samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex2
Dynamic Array:
Data: [10, 20, 30]
Size: 3
Capacity: 5
```

b) Access by index

```
• samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex2
Dynamic Array:
Data: [10, 20, 30]
Size: 3
Capacity: 5

At Index 2: 30
```

c) Resize

```
• samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex2
Dynamic Array:
Data: [10, 20, 30]
Size: 3
Capacity: 5

At Index 2: 30

After resizing:
Dynamic Array:
Data: [10, 20]
Size: 2
Capacity: 2
```

Advanced Operations

a) Insertions

```
After insertions:
Dynamic Array:
Data: [5, 4, 10, 20, 3]
Size: 5
Capacity: 8
```

b) Removals

```
After insertions:
Dynamic Array:
Data: [5, 4, 10, 20, 3]
Size: 5
Capacity: 8

After removals:
Dynamic Array:
Data: [4, 20]
Size: 2
Capacity: 8
```

c) Finding

```
samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex2
Dynamic Array:
Data: [10, 20, 30]
Size: 3
Capacity: 5
At Index 2: 30
```

d) Clearing

```
After clearing the array:
Dynamic Array:
Data: []
Size: 0
Capacity: 8
```

Edge Cases

a) Empty Array

```
samirvarma@Samirs-MacBook-Pro-2 Lab7 % gcc ex2.c -o ex2
samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex2
Dynamic Array:
Data: []
Size: 0
Capacity: 5
```

b) Single Element

```
Dynamic Array:
Data: [1]
Size: 1
Capacity: 5
```

c) Trigger Resize

```
samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex2
Dynamic Array:
Data: []
Size: 0
Capacity: 2

Dynamic Array:
Data: [1, 1, 1]
Size: 3
Capacity: 4
```

d) Negative size/Null input

```
samirvarma@Samirs-MacBook-Pro-2 Lab7 % gcc ex2.c -o ex2
samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex2
Invalid array.

After clearing the array:
Invalid array.
samirvarma@Samirs-MacBook-Pro-2 Lab7 %
```

Exercise 3

Basic Operations

- a) Create list, add to start/end

```
Linked List:  
Head -> [10] -> [20] -> [30] -> [40] -> NULL  
Size: 4
```

- b) Clearing list

```
After clearing the list:  
List is empty.
```

Advanced Operations

- a) Insert, remove, find elements at specific positions

```
● samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex3  
Linked List:  
Head -> [10] -> [20] -> [30] -> [40] -> NULL  
Size: 4  
  
After inserting 25 at position 2:  
Linked List:  
Head -> [10] -> [20] -> [25] -> [30] -> [40] -> NULL  
Size: 5  
  
After removing element at position 3:  
Linked List:  
Head -> [10] -> [20] -> [25] -> [40] -> NULL  
Size: 4  
  
Element at position 2: 25  
Position of value 40: 3
```

Edge Cases

- a) Empty Lists

```
● samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex3
List is empty.
Element at position 2: -1
Position of value 40: -1
After clearing the list:
List is empty.
```

b) Single element list

```
● samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex3
Linked List:
Head -> [20] -> NULL
Size: 1

After clearing the list:
List is empty.
○ samirvarma@Samirs-MacBook-Pro-2 Lab7 %
```

c) Invalid positions

```
● samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex3
Linked List:
Head -> [20] -> NULL
Size: 1

Element at position 2: -1
Position of value 40: -1
After clearing the list:
List is empty.
```

d) NULL inputs

```
● samirvarma@Samirs-MacBook-Pro-2 Lab7 % ./ex3
Testing NULL input handling:
Add First (NULL list): 0
Add Last (NULL list): 0
Insert At (NULL list): 0
Remove At (NULL list): 0
Get At (NULL list): -1
```