TEST CASES - DISJOINT SET

CHANGES MADE

Scenario 1:

Inserting a negative number

Previous:

No restrictions placed on negative numbers

After changes:

Negative numbers are not allowed for insertion

Output:

"Negative numbers cannot be inserted. Failed to create set"

Scenario 2:

Repeated insertion/ Insertion of an pre-existant element

Previous:

Though insertion of same element repeatedly was not allowed, there was newnode creation leading to wastage of space

After changes:

Element already exists is displayed, 'newnode' is freed

Output:

"Element is already in set. Failed to create set"

Scenario 3:

Prompt change to avoid excess number of entries

Previous:

Enter elements to merge was displayed as prompt

After changes:

Enter 2 elements to merge is displayed as prompt

Output:

“Enter 2 elements of any sets to be merged”

Scenario 4:

Finding elements greater than ‘capacity’ without inserting

Previous:
Program would terminate due to segmentation fault if we find element greater than ‘capacity’ without inserting

After changes:

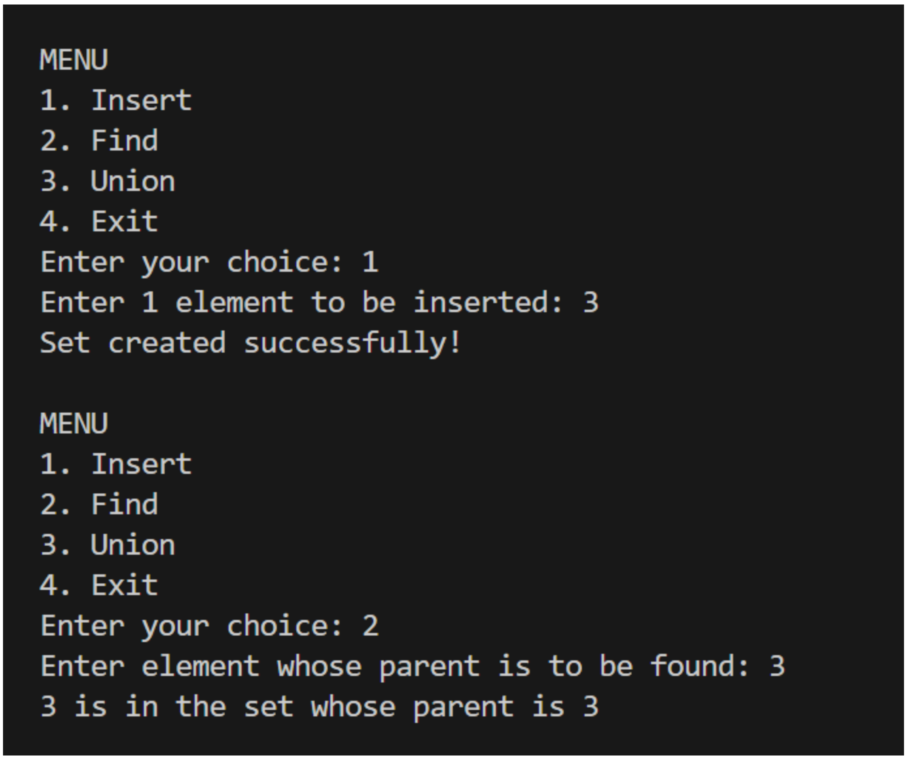
Element that is not inserted is not found

Output:

“Element not found.”

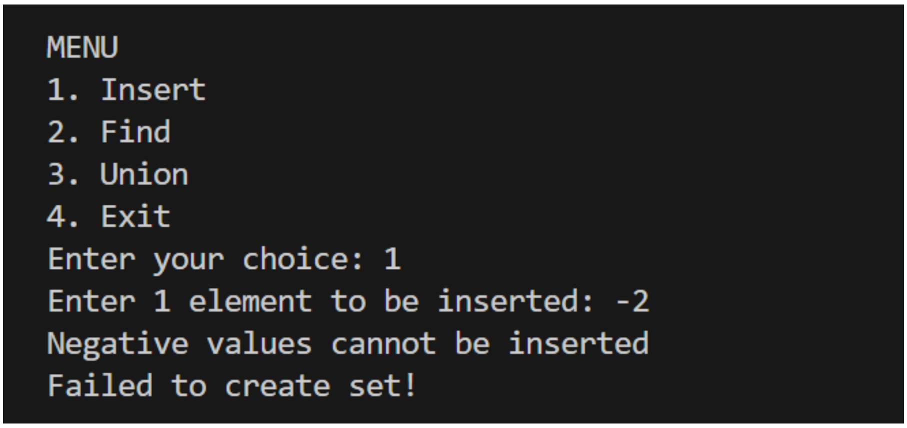
OUTPUT OF POSSIBLE TEST CASES

1. **Inserting an element in an empty set and finding it**



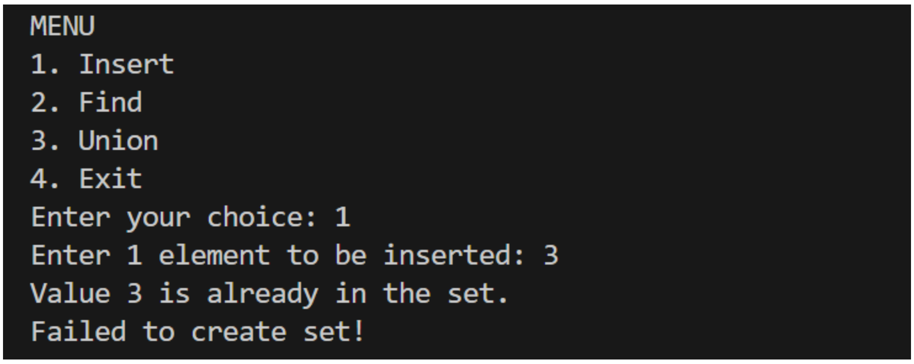
```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 1
Enter 1 element to be inserted: 3
Set created successfully!

MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 3
3 is in the set whose parent is 3
```

2. **Blocking the insertion of a negative elements**



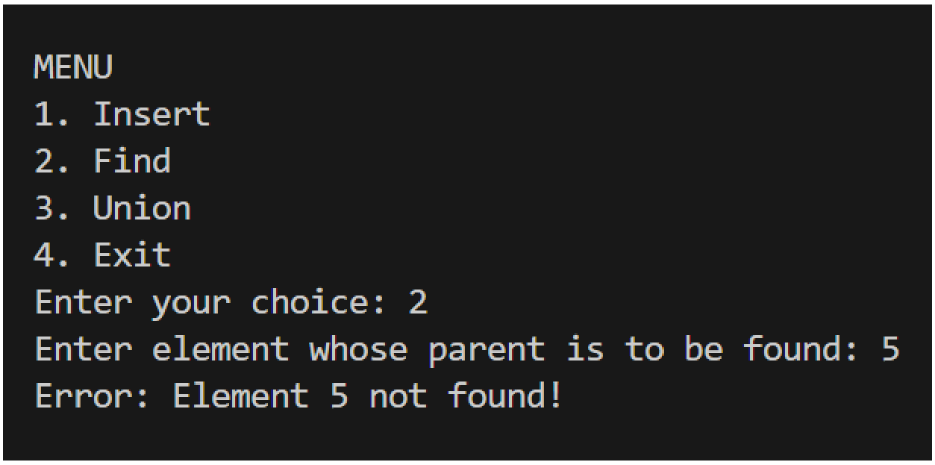
```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 1
Enter 1 element to be inserted: -2
Negative values cannot be inserted
Failed to create set!
```

3. **Blocking the insertion of an already existant element**



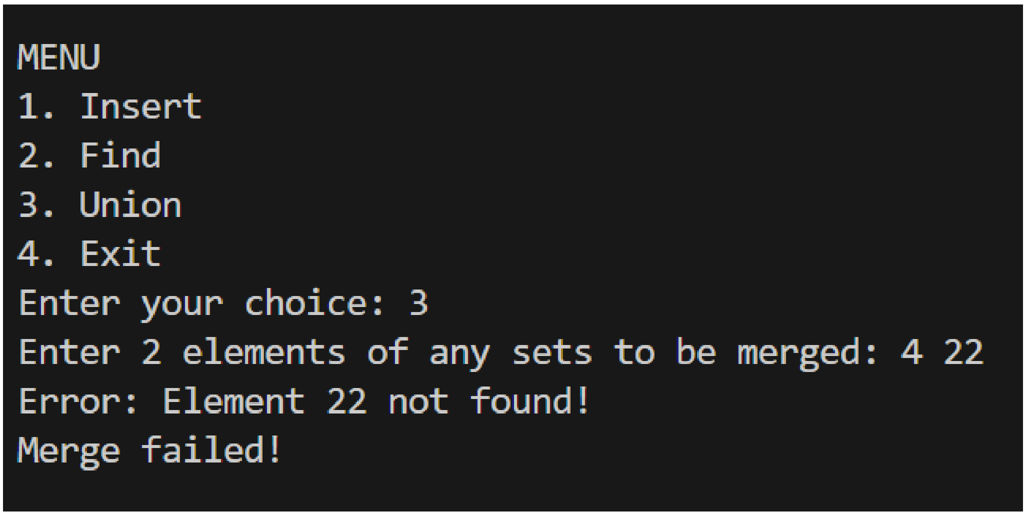
```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 1
Enter 1 element to be inserted: 3
Value 3 is already in the set.
Failed to create set!
```

4. **Finding non-existant element**



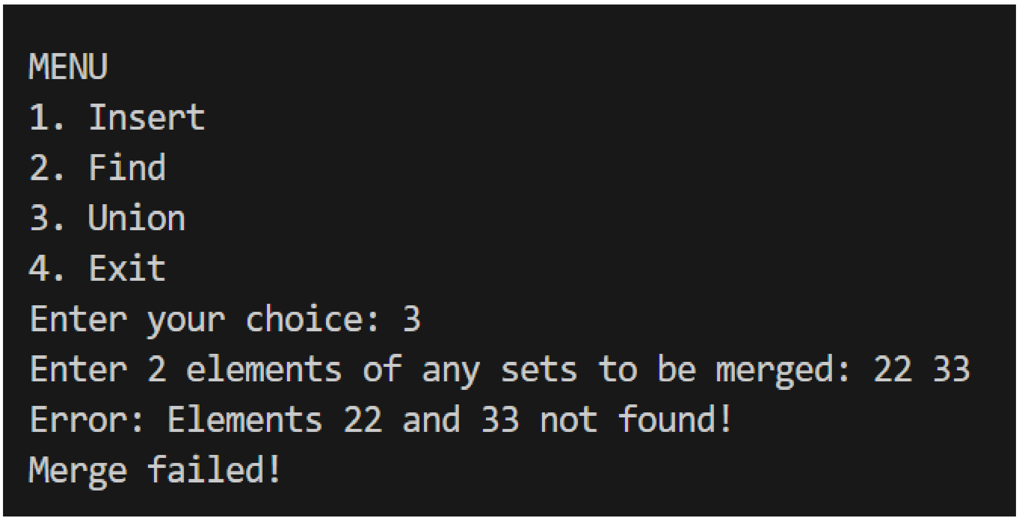
```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 5
Error: Element 5 not found!
```

5. **Merging an element with a non-existinant element**



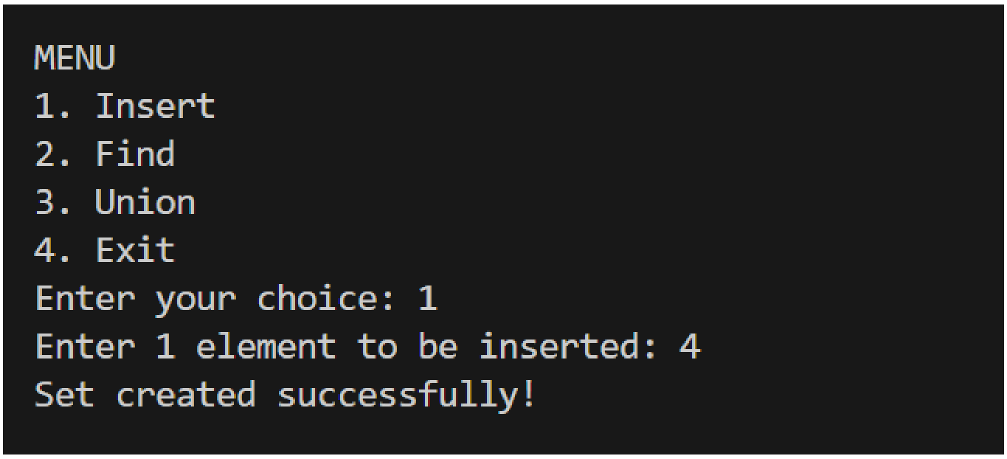
```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 3
Enter 2 elements of any sets to be merged: 4 22
Error: Element 22 not found!
Merge failed!
```

6. **Merging two non-existant elements**

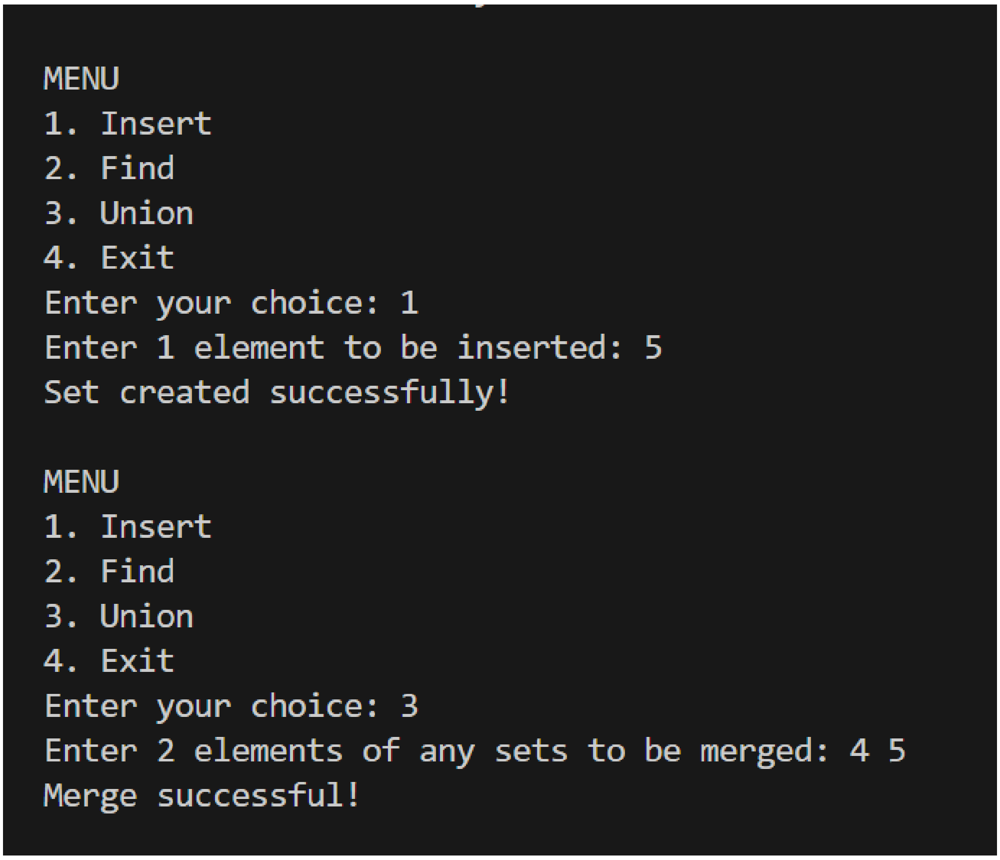


```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 3
Enter 2 elements of any sets to be merged: 22 33
Error: Elements 22 and 33 not found!
Merge failed!
```

7. **Inserting two new elements, and merging the two disjoint-sets**

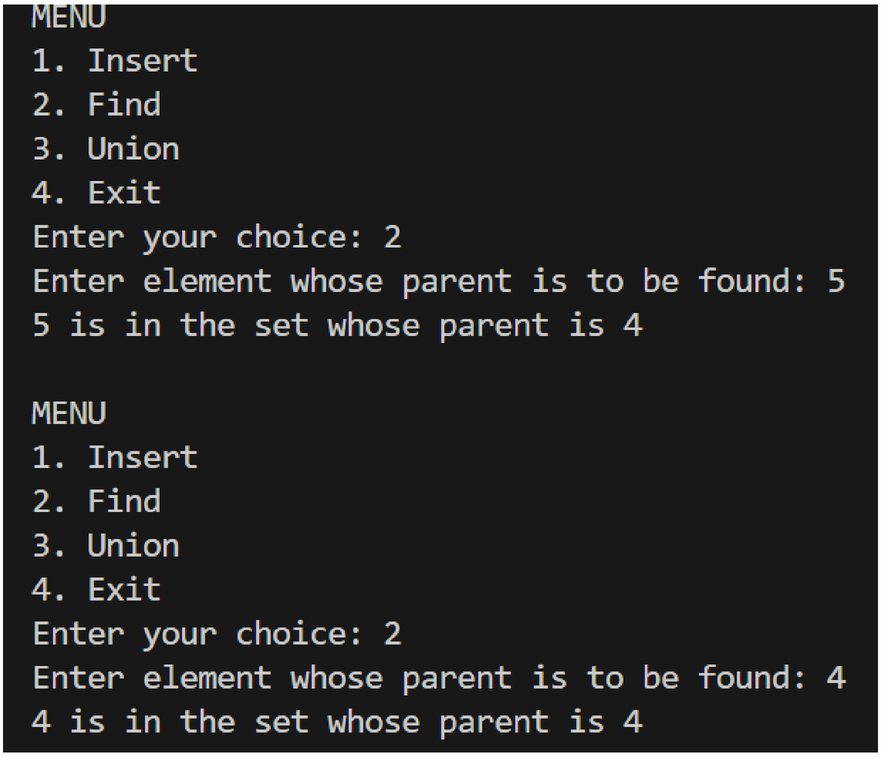


```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 1
Enter 1 element to be inserted: 4
Set created successfully!
```



```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 1
Enter 1 element to be inserted: 5
Set created successfully!

MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 3
Enter 2 elements of any sets to be merged: 4 5
Merge successful!
```

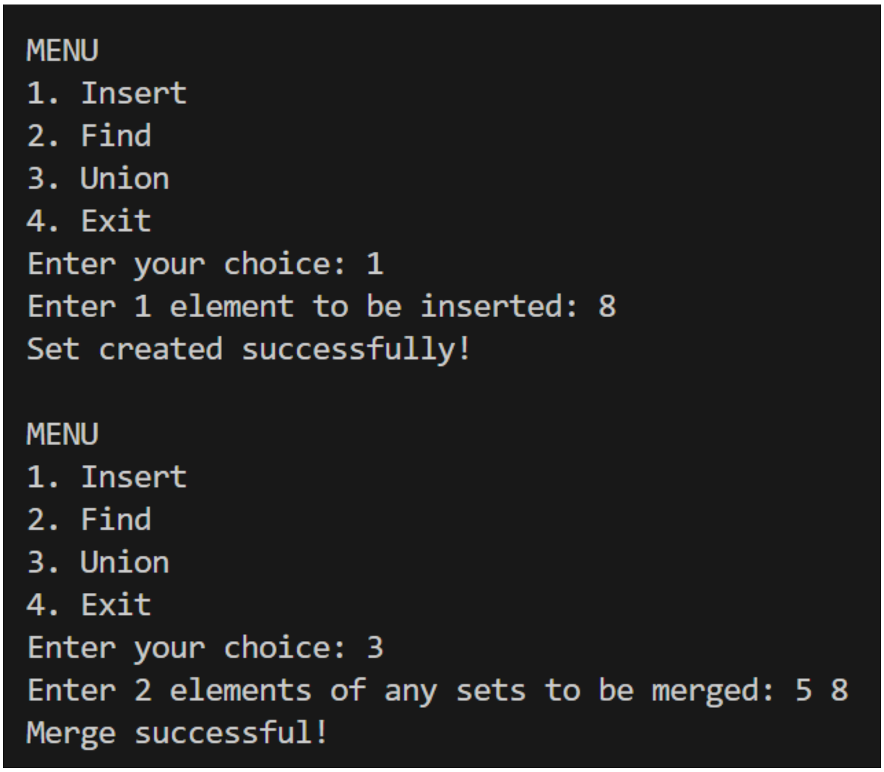
8. **Finding element in a merged set**



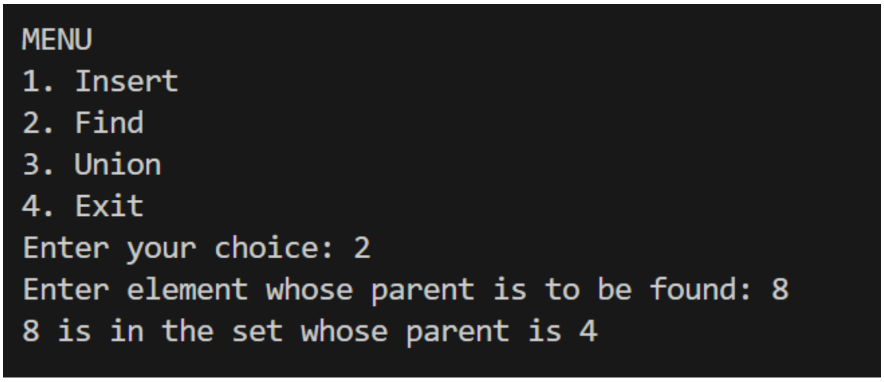
```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 5
5 is in the set whose parent is 4

MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 4
4 is in the set whose parent is 4
```

9. Creating a more complex tree:

9.1. Merging single-element set with higher rank disjoint set and finding an element in the merged set

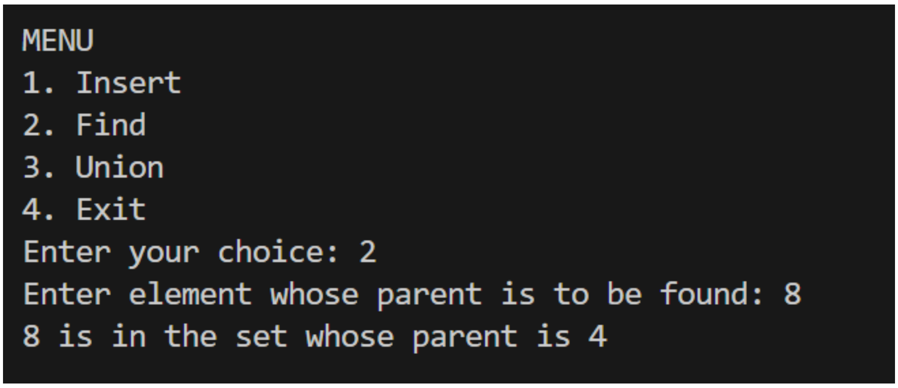


```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 1
Enter 1 element to be inserted: 8
Set created successfully!

MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 3
Enter 2 elements of any sets to be merged: 5 8
Merge successful!
```

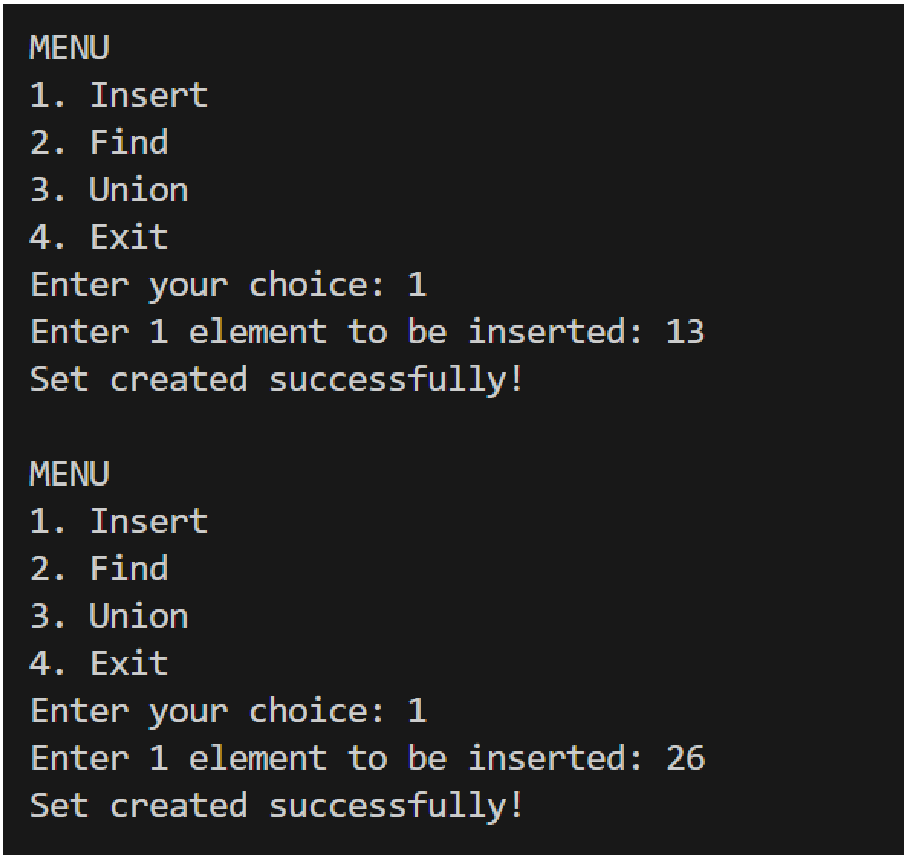


```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 8
8 is in the set whose parent is 4
```

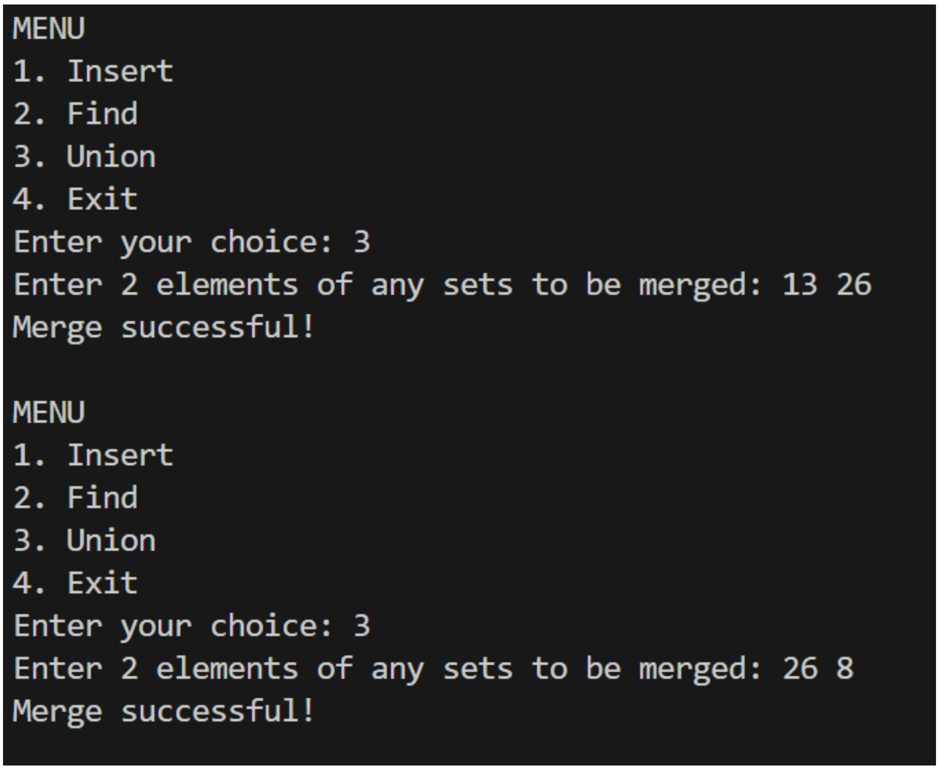
9.2. Merging multi-element sets of different ranks and finding an element in the merged set



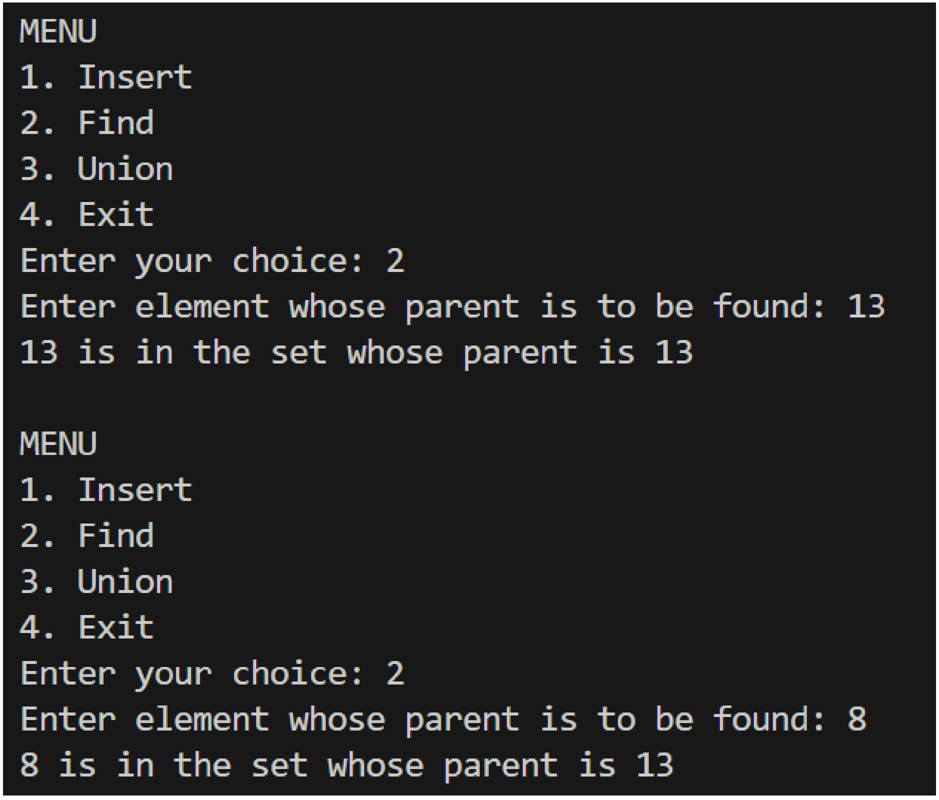
```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 8
8 is in the set whose parent is 4
```



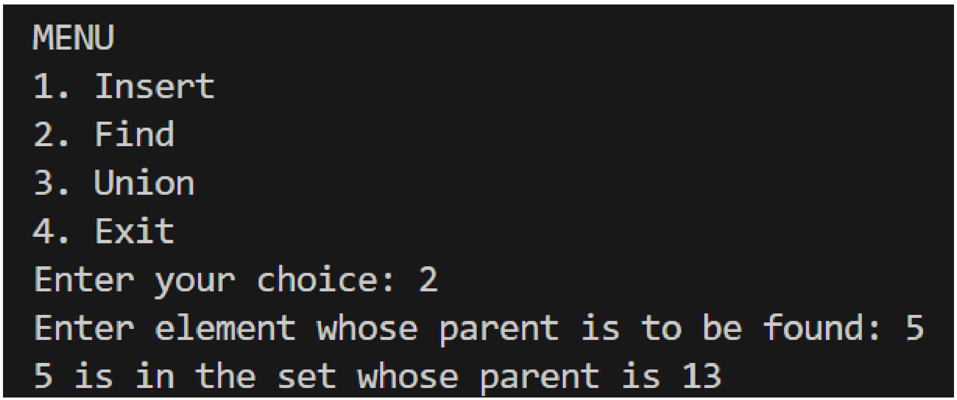
```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 1
Enter 1 element to be inserted: 13
Set created successfully!

MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 1
Enter 1 element to be inserted: 26
Set created successfully!
```



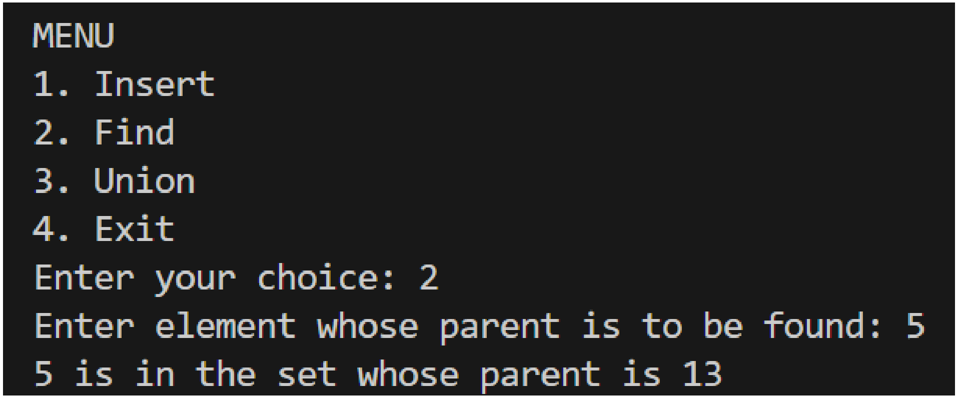
```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 3
Enter 2 elements of any sets to be merged: 13 26
Merge successful!

MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 3
Enter 2 elements of any sets to be merged: 26 8
Merge successful!
```



```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 13
13 is in the set whose parent is 13

MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 8
8 is in the set whose parent is 13
```

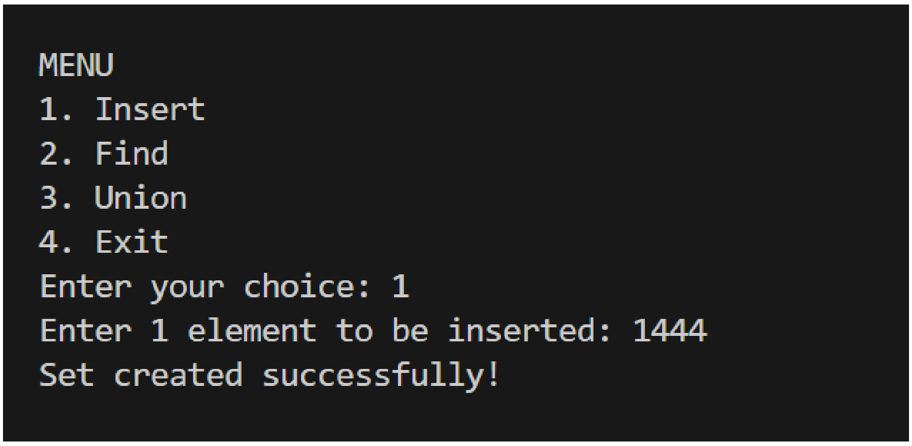


```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 5
5 is in the set whose parent is 13
```



```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 2
Enter element whose parent is to be found: 5
5 is in the set whose parent is 13
```

10. Inserting a number greater than current capacity



```
MENU
1. Insert
2. Find
3. Union
4. Exit
Enter your choice: 1
Enter 1 element to be inserted: 1444
Set created successfully!
```

SUMMARY

Function name	Pre-condition	Expected Result	Status
INSERT	Non-existent integer	Insertion successful	Passed
	Existent integer	Insertion fails	Passed
	Negative integer	Insertion fails	Passed
	Number greater than 'capacity'	Insertion successful	Passed
FIND	Existent single element	Found (Displays the parent)	Passed
	Non-existent single element	Element not found	Passed
MERGE	2 Existent elements	Merged successfully	Passed
	1 Existent and 1 Non-existent element	Element not found	Passed
	2 Non-existent elements	Both elements not found	Passed
	2 Sets with multiple elements	Merged successfully	Passed