



جامعة الإسكندرية  
ALEXANDRIA  
UNIVERSITY  
كلية الهندسة  
Faculty of Engineering



# Programming I Project

## Bank Management System

*Omar Ahmed Abdeldayem [8764]*  
*Mootassem Emad El-Zarka [8586]*  
*Raphael Elkess Bichoy [8683]*  
*Abdelrahman Essam Mohamed [9053]*

## **Description of the features of the system**

### **Feature Add New Account:**

The "Add New Account" feature of the Bank Management System allows the creation of a new customer account with specific details. This feature enhances the system's functionality by enabling users to efficiently input accurate information for the creation of a new account.

#### 1. User Input:

- Users initiate the process by selecting the "Add New Account" option from the system menu.
- The system prompts the user to enter essential details, including name, mobile number, and email.

#### 2. Name Validation:

- The system checks the format of the entered name to ensure it adheres to valid naming conventions.
- If an invalid name format is detected, users are prompted to correct the input.

#### 3. Mobile Validation:

- Similar to the name, the system validates the mobile number format to ensure accuracy.
- Users have the option to correct the mobile number if an error is detected.

#### 4. Email Validation:

- The system validates the email format to ensure it follows standard email conventions.
- Users are prompted to correct the email input if an invalid format is detected.

#### 5. Account Number Generation:

- A unique account number is generated for the new account by incrementing the maximum account number by a constant value.
- This ensures the uniqueness of each account number in the system.

#### 6. Date Initialization:

- The system automatically records the date of account creation using the current system date.

#### 7. Account Creation:

- Upon successful validation of user inputs, a new account is created with the provided details.
- The account is added to the system's array of accounts, and the array is dynamically resized to accommodate the new entry.

#### 8. Data Persistence:

- The system saves the newly created account details to maintain data consistency.
- If the save operation is successful, a confirmation message is displayed.

This feature simplifies the process of adding new accounts to the bank system, ensuring that user inputs are accurate and adhere to predefined formats. The validation checks contribute to data integrity, and the dynamic resizing of the account array accommodates the growing number of customer accounts.

## **Feature Delete Existing Account:**

The "Delete Existing Account" feature of the Bank Management System enables users to remove an account from the system under specific conditions. This feature is designed to maintain data integrity by restricting the deletion of accounts with non-zero balances.

### **1. User Input:**

- Users initiate the deletion process by selecting the "Delete Existing Account" option from the system menu.
- The system prompts users to enter the account number associated with the account to be deleted.

### **2. Account Number Validation:**

- The entered account number is validated to ensure it exists in the system.
- If an invalid account number is detected, users are notified, and the deletion process is halted.

### **3. Balance Check:**

- The system checks if the account to be deleted has a balance greater than or equal to 0.01.
- If the balance is not zero, the deletion process is aborted to prevent the removal of accounts with remaining funds.

### **4. Data Exchange:**

- If the account passes the validation checks, its data is exchanged with the last account in the system's array.
- This exchange ensures that the account to be deleted is replaced by the last account in the array.

### **5. Array Update:**

- The system decreases the total number of accounts by one to reflect the successful deletion.
- The array of accounts is dynamically resized to remove the last account, effectively deleting the specified account.

### **6. Data Persistence:**

- If the deletion is successful, the system saves the updated account data to maintain consistency.
- Additionally, the corresponding account file is removed from the file system.

This feature provides a controlled method for removing accounts, emphasizing the importance of maintaining a non-zero balance for account deletion. The data exchange mechanism ensures that the deleted account is replaced by an existing one, preventing any disruption to the structure of the account array.

## **Feature Modify Account Information:**

The "Modify Account Information" feature in the Bank Management System empowers users to update specific details associated with an existing account. This functionality enhances the system's flexibility by allowing users to adapt their account information to reflect changes accurately.

### **1. User Input:**

- Users begin the modification process by selecting the "Modify Account Information" option from the system menu.
- The system prompts users to enter the account number for the account they wish to modify.

### **2. Account Number Validation:**

- The entered account number undergoes validation to ensure its existence in the system.
- If an invalid account number is detected, users are notified, and the modification process is terminated.

### **3. Modification Options:**

- Users are presented with a menu offering different aspects of the account information that can be modified.

### **4. User Selection:**

- Users choose the specific aspect they want to modify by entering the corresponding option.
- Depending on the chosen option, users are prompted to enter the new information.

### **5. Input Validation:**

- The system validates the new information to ensure it adheres to the required format (e.g., valid name, mobile number, or email address).
- If the input is invalid, users are informed, and they can re-enter the information.

### **6. Data Update:**

- If the input is valid, the system updates the corresponding account information with the new data.

### **7. Menu Loop:**

- After modifying one aspect, users have the option to modify another or exit the modification menu.

### **8. Data Persistence:**

- Upon completion of the modification process, the system saves the updated account information to maintain consistency.

This feature provides users with the ability to keep their account information up to date, ensuring that the system reflects accurate and current details. The modular approach allows users to selectively modify specific aspects based on their needs.

## **Feature Search for Account:**

The "Search for Account" feature in the Bank Management System provides users with the capability to locate specific account details based on the entered account number. This functionality enhances user experience by offering a quick and convenient method to retrieve account information.

### **1. User Input:**

- Users initiate the search process by selecting the "Search for Account" option from the system menu.
- The system prompts users to enter the account number they want to search for.

### **2. Account Number Validation:**

- The entered account number undergoes validation to ensure its existence in the system.
- If an invalid account number is detected, users are notified, and the search process is terminated.

### **3. Search Operation:**

- The system performs a search through the existing accounts to find a match with the entered account number.

### **4. Result Display:**

- If a matching account is found, the system displays the detailed information associated with that account.
- Information includes the account number, account holder's name, mobile number, email address, balance, and the date the account was opened.

### **5. Not Found Message:**

- If no matching account is found, users are informed that the account number was not found.
- A suggestion is provided to double-check the entered account number and try again.

### **6. Data Security:**

- The feature ensures that account numbers are validated to prevent unauthorized access to account information.

This feature is designed to save time for both users and system administrators by providing a straightforward method to access specific account details without the need to navigate through extensive menus.

## **Feature Advanced Search for Accounts:**

The "Advanced Search for Accounts" feature in the Bank Management System provides users with an enhanced search capability based on a user-defined keyword. This feature offers flexibility for users who want to search for accounts containing specific information, such as account holder names, matching a given keyword.

### **1. User Input:**

- Users initiate the advanced search by selecting the "Advanced Search for Accounts" option from the system menu.
- The system prompts users to enter a keyword that they want to use for the search.

### **2. Search Operation:**

- The system scans through the existing accounts to find matches with the entered keyword.
- The search is applied to account holder names to identify relevant accounts considering the name with any lowercases or uppercases.

### **3. Result Display:**

- If accounts matching the keyword are found, the system displays detailed information for each matching account.
- Information includes the account number, account holder's name, mobile number, email address, balance, and the date the account was opened.

### **4. No Match Found Message:**

- If no matching accounts are found, users are informed that no accounts were found based on the entered keyword.
- Users can modify the keyword and retry the search.

### **5. Flexible Search Criteria:**

- The feature provides flexibility by allowing users to define their search criteria based on keywords, making it easier to locate specific accounts.

### **6. Efficient Data Retrieval:**

- This feature enhances user efficiency by quickly providing relevant account details, especially in scenarios where users want to search based on specific names.

This feature contributes to an improved user experience by empowering users with a flexible and efficient search mechanism for accounts based on user-defined criteria.

## **Feature Withdrawal Operation:**

The "Withdrawal Operation" feature in the Bank Management System allows account holders to make withdrawals from their accounts. This feature ensures secure and accurate transaction processing while maintaining the integrity of account balances.

### **1. User Input:**

- Users initiate a withdrawal by selecting the "Withdrawal" option from the system menu.
- The system prompts users to enter their account number for identification.

### **2. Account Validation:**

- The system validates the entered account number to ensure its accuracy and existence in the system.
- If the account number is invalid or not found, an error message is displayed, and the transaction is terminated.

### **3. Withdrawal Amount Input:**

- Users are prompted to enter the withdrawal amount, with a maximum limit of \$10,000.
- The system validates the entered amount to ensure it is in the correct format and adheres to the specified limit.

### **4. Validation Checks:**

- The system checks whether the withdrawal amount is greater than zero and does not exceed the account balance.
- If the amount is invalid or exceeds the account balance, appropriate error messages are displayed.

### **5. Transaction Processing:**

- If all validation checks pass, the withdrawal operation is performed by deducting the specified amount from the account balance.

### **6. Transaction Logging:**

- The system logs the withdrawal transaction, recording details such as the withdrawn amount and the updated account balance.
- A transaction log file is maintained for each account, ensuring a comprehensive record of all transactions.

### **7. User Notification:**

- After successful completion of the withdrawal, the system provides users with a confirmation message.
- The message includes details such as the withdrawal amount and the new account balance.

### **8. Data Persistence:**

- The system ensures data persistence by saving the updated account information, including the new balance, to the corresponding account file.

### **9. Error Handling:**

- The system includes error-handling mechanisms to inform users of any issues during the withdrawal process, such as insufficient balance.

### **10. Enhanced Security:**

- The withdrawal feature incorporates security measures to protect user accounts and transactions, preventing unauthorized access.

This feature streamlines the withdrawal process for account holders, providing a secure and user-friendly experience while maintaining accurate financial records.

## **Feature Deposit Operation:**

The "Deposit Operation" feature in the Bank Management System empowers account holders to add funds to their accounts securely and efficiently. This feature ensures accuracy in transaction processing while maintaining a comprehensive record of financial activities.

### **1. User Input:**

- Users initiate a deposit by selecting the "Deposit" option from the system menu.
- The system prompts users to enter their account number for identification.

### **2. Account Validation:**

- The system validates the entered account number to ensure its accuracy and existence in the system.
- If the account number is invalid or not found, an error message is displayed, and the transaction is terminated.

### **3. Deposit Amount Input:**

- Users are prompted to enter the withdrawal amount, with a maximum limit of \$10,000.
- The system validates the entered amount to ensure it is in the correct format and adheres to the specified limit.

### **4. Validation Checks:**

- The system checks whether the withdrawal amount is greater than zero.
- If the amount is invalid or exceeds the account balance, appropriate error messages are displayed.

### **5. Transaction Processing:**

- If all validation checks pass, the deposit operation is performed by adding the specified amount to the account balance.

### **6. Transaction Logging:**

- The system logs the deposit transaction, recording details such as the deposited amount and the updated account balance.
- A transaction log file is maintained for each account, providing a detailed history of financial activities.

### **7. User Notification:**

- After successful completion of the deposit, the system provides users with a confirmation message.
- The message includes details such as the deposited amount and the new account balance.

### **8. Data Persistence:**

- The system ensures data persistence by saving the updated account information, including the new balance, to the corresponding account file.

### **9. Error Handling:**

- The system includes error-handling mechanisms to inform users of any issues during the deposit process.

### **10. Enhanced Security:**

- The withdrawal feature incorporates security measures to protect user accounts and transactions, preventing unauthorized access.

This feature simplifies the process of adding funds to an account, offering account holders a convenient and secure means of managing their finances.



## **Feature Transfer Operation:**

The "Transfer Operation" feature in the Bank Management System allows users to perform secure and efficient fund transfers between different accounts. This operation ensures accurate and recorded movement of funds between the sender and the receiver accounts.

### **1. Sender Account Input:**

- Users are prompted to enter the account number from which they want to initiate the transfer.

### **2. Sender Account Validation:**

- The system validates the sender's account number to ensure its accuracy and existence.
- If the account is not found or is invalid, an error message is displayed, and the transfer is terminated.

### **3. Receiver Account Input:**

- Users input the account number of the recipient.

### **4. Receiver Account Validation:**

- The system validates the receiver's account number for accuracy and existence.
- If the receiver's account is not found or is invalid, an error message is displayed, and the transfer is rejected.

### **5. Same Account Check:**

- The system checks if the sender and receiver account numbers are the same.
- If identical, an error message is displayed, and the transfer is not allowed.

### **6. Transfer Amount Input:**

- Users specify the amount of funds they intend to transfer.

### **7. Amount Validation:**

- The system validates the specified transfer amount, ensuring it meets required format criteria.
- If the amount is invalid, an error message is displayed, and the transfer is declined.

### **8. Sufficient Balance Check:**

- The system checks if the sender's account has sufficient funds to cover the transfer amount.
- If the balance is insufficient, an error message is displayed, and the transfer is not allowed.

### **9. Transfer Execution:**

- If all validations pass, the system deducts the transfer amount from the sender's account and adds it to the receiver's account.

### **10. Transaction Logging:**

- The system records the transfer transactions for both the sender and the receiver, detailing the transferred amount and updated balances.

### **11. User Notification:**

- Users receive confirmation messages indicating the successful completion of the transfer and providing the updated balances of both accounts.

#### 12. Data Persistence:

- The system ensures data persistence by saving the updated account information, including the new balances, to the corresponding account files.

#### 13. Error Handling:

- Robust error-handling mechanisms are implemented to inform users of any issues during the transfer operation.

#### 14. Security Measures:

- Security protocols are in place to authenticate users and safeguard against unauthorized fund transfers.

The "Transfer Operation" feature enhances the functionality of the Bank Management System, offering users a reliable means to transfer funds securely between accounts.

### **Feature Account Report:**

The "Account Report" feature in the Bank Management System allows users to generate a concise report summarizing the recent transactions for a specific account. This report provides users with insights into the account's financial activities, ensuring transparency and accountability.

#### 1. Account Number Input:

- Users are prompted to enter the account number for which they want to generate the report.

#### 2. Account Number Validation:

- The system validates the entered account number to ensure its accuracy and existence.
- If the account is not found or is invalid, an error message is displayed, and the report generation is terminated.

#### 3. File Access:

- The system attempts to open the corresponding account file for reading.

#### 4. File Existence Check:

- If the file does not exist, an error message is displayed, and the report generation is halted.

#### 5. Transaction Count:

- The system counts the total number of transactions recorded in the account file.

#### 6. Transaction Retrieval:

- If transactions are present, the system retrieves the last five transactions from the account file.
- If there are fewer than five transactions, all transactions are retrieved.

#### 7. Report Display:

- The system displays the retrieved transactions, including details such as transaction type, amount, and updated balance.
- For brevity, only the most recent transactions are displayed.

#### 8. Error Handling:

- Robust error-handling mechanisms are implemented to inform users of any issues during the report generation.

The "Account Report" feature enhances the user experience by providing a quick overview of the recent financial activities associated with a specific account.

## **Feature Print Sorted Data:**

The "Print Sorted Data" feature in the Bank Management System enables users to print a summary of account information based on their preferred sorting order. Users can choose to display the data either in a tabular format or as a simple list.

### **1. Sorting Order Selection:**

- Users are prompted to choose the sorting order for printing the data.
- Options include sorting by name, balance, or open date.

### **2. Input Validation:**

- The system validates user input to ensure a correct sorting order selection.

### **3. Data Sorting:**

- The system utilizes the selected sorting order to arrange the account data accordingly.
- Sorting is performed by names (A-Z), balance (ascending), or date (oldest to newest).

### **4. Table Format Confirmation:**

- Users are given the option to choose between displaying the data in a table format or a simple list.
- The system validates the input for this option.

### **5. Print Information:**

- The system prints a header indicating the chosen sorting order.
- If the user opts for the table format, a formatted table is printed.
- If the user prefers a simple list, individual account details are printed.

The "Print Sorted Data" feature offers flexibility in presenting account information, allowing users to customize the display based on their preferences.

## User Manuel with Sample Runs

### Notes:

- Any function that will edit account details; you will have to save.
- Validate data for functions requiring input; errors will prompt correction.

### Display:

1. Login: *Enter username and password.*
2. Quit: *Exit the program.*

```
                                Welcome to Bank Management System
Commands:
    1. Login
    2. Quit
>>>
```

**After Successful Login:** Display the menu; retry login if credentials are incorrect.

### List of options appears:

```
                                Hello soma
=====
Commands:
    1. Add           2. Delete           3. Modify
    4. Search        5. Advanced Search  6. Withdraw
    7. Deposit       8. Transfer          9. Report
    10. Print        11. Quit
>>> |
```

## Menu Options:

### I. Add Account (*Press 1*):

- Enter new account data: name, email and mobile.
- Account number is automatically generated.
- Date opened auto filled with the current system date.
- The balance for the new account is 0 by default.
- Press 1 to save the account or 0 to discard the added account.

```
>>> 1
Enter new account details:

Enter name: Micheal Jones
Enter mobile: 01009700000
Enter email: m.jones@gmail.com
type 1 to save date or type zero to discard changes: 1
Data Saved

New Account is added successfully.
=====
```

### II. Delete Account (*Press 2*):

- Enter a valid account number; deletion requires zero balance.

```
>>> 2
please enter account number for account to be deleted: 4444455654
type 1 to save date or type zero to discard changes: 1
Data Saved

=====
```

### III. Modify Account (*Press 3*):

- Enter account number.
- Press 1 to modify name.
- Press 2 to modify mobile
- Press 3 to modify email address
- Press 0 to exit then 1 to save modification or 0 to discard

- Multiple modifications are allowed

```
>>> 3
please enter the account number for the account: 4444455555

Type 1 to modify name, 2 to modify mobile, 3 to modify email address and 0 to exit
>>> 1
enter the new name: Micheal Jones

Type 1 to modify name, 2 to modify mobile, 3 to modify email address and 0 to exit
>>> 2
enter the new mobile number: 01009700000

Type 1 to modify name, 2 to modify mobile, 3 to modify email address and 0 to exit
>>> 3
enter the new email address: m.jones@gmail.com

Type 1 to modify name, 2 to modify mobile, 3 to modify email address and 0 to exit
>>> 0
type 1 to save date or type zero to discard changes: 1
Data Saved

=====
```

#### IV. Search Account (*Press 4*):

- Enter the account number for the desired account data to be printed.

```
>>> 4
please enter the account number for the account: 4444455555
Account Number: 4444455555
Name: Micheal Jones
E-mail: m.jones@gmail.com
Balance: 480789.13 $
Mobile: 01009700000
Opened: May 2023

=====
```

V. Advanced Search (*Press 5*):

- Enter a keyword to search for all the accounts names that contain the desired keyword.

```
>>> 5
Enter your keyword: Micheal
Account Number: 4444455555
Name: Micheal Jones
E-mail: m.jones@gmail.com
Balance: 480789.13 $
Mobile: 01009700000
Opened: May 2023
```

```
Account Number: 9700000003
Name: Micheal Robert
E-mail: michael@yahoo.com
Balance: 300.00 $
Mobile: 01009700003
Opened: November 2008
```

```
=====
```

VI. Withdraw (*Press 6*):

- Enter account number for withdrawal.
- Enter withdrawal amount.
- Balance must exceed withdrawal amount
- Max withdrawal amount is 10,000 \$.
- Press 1 to proceed the transaction or 0 to discard.

```
>>> 6
Enter account number for withdrawal: 3333344444
Enter withdrawal amount (max 10,000 $): 750
type 1 to save date or type zero to discard changes: 1
Data Saved
```

```
Withdrawal successful. New balance: 605639.00 $
```

```
=====
```

## VII. Deposit (*Press 7*):

- Enter account number for deposit.
- Enter deposit amount.
- Max deposit amount is 10,000 \$.
- Press 1 to proceed the transaction or 0 to discard.

```
>>> 7
Enter account number for deposit: 1111122222
Enter deposit amount (max 10,000 $): 8007.985
type 1 to save date or type zero to discard changes: 1
Data Saved

Deposit successful. New balance: 19558.46 $
=====
```

## VIII. Transfer (*Press 8*):

- Enter sender's account number and receiver's account number. (They must be different)
- Enter transfer amount.
- Sender's balance must exceed transfer amount.
- Press 1 to proceed the transaction or 0 to discard.

```
>>> 8
Enter sender account number: 3333344444
Enter receiver account number: 1111122222
Enter transfer amount: 5463
type 1 to save date or type zero to discard changes: 1
Data Saved

Transfer successful. New balance in sender's account: 600176.00 $
Transfer successful. New balance in receiver's account: 25021.46 $
=====
```



## IX. Print Report (*Press 9*):

- Enter account number to print the last 5 transactions of this account.

```
>>> 9
Enter the account number: 3333344444
Deposit:5876.90, Balance:606389.00 $
Withdraw:750.00, Balance:605639.00 $
Transfer (Outgoing):5463.00, Balance:600176.00 $
Transfer (Incoming):1250.00, Balance:601426.00 $
Withdraw:9999.00, Balance:591427.00 $
=====
```

## X. Print Sorted Data (*Press 10*):

- Press 1 for sort by name (A to Z).
- Press 2 for sort by balance (Ascendingly).
- Press 3 for sort by open date (Old to New).
- For table format print Press 1 else Press 0.

```
>>> 10
Could you please specify the sorting order you prefer for the data to be printed?
1 for sorted by name / 2 for sorted by balance / 3 for sorted by open date
>>> 1
Do you want to print it in table format? yes:1 no:0 1
Sorted by names(a-z)
Number | Name | Email | Balance | Mobile | Month | Year
-----|-----|-----|-----|-----|-----|-----
9700000102 | Abdelrahman Essam | abdelrahman@yahoo.com | 0.00 | 01227894265 | December | 2023
4444455555 | Micheal Jones | m.jones@gmail.com | 480789.13 | 01009700000 | May | 2023
9700000003 | Micheal Robert | michael@yahoo.com | 300.00 | 01009700003 | November | 2008
3333344444 | Mootassem Elzarka | mootassem@gmail.com | 591427.00 | 01234567899 | May | 2019
2222233333 | Omar Hekal | omar@gmail.com | 40000.73 | 01122553164 | December | 2015
1111122222 | Raphy Takla | raphy@gmail.com | 23771.46 | 01278967897 | September | 2011
=====
```

```
>>> 10
Could you please specify the sorting order you prefer for the data to be printed?
1 for sorted by name / 2 for sorted by balance / 3 for sorted by open date
>>> 2
Do you want to print it in table format? yes:1 no:0 1
Sorted by Balance (ascendingly)
Number | Name | Email | Balance | Mobile | Month | Year
-----|-----|-----|-----|-----|-----|-----
9700000102 | Abdelrahman Essam | abdelrahman@yahoo.com | 0.00 | 01227894265 | December | 2023
9700000003 | Micheal Robert | michael@yahoo.com | 300.00 | 01009700003 | November | 2008
1111122222 | Raphy Takla | raphy@gmail.com | 23771.46 | 01278967897 | September | 2011
2222233333 | Omar Hekal | omar@gmail.com | 40000.73 | 01122553164 | December | 2015
4444455555 | Micheal Jones | m.jones@gmail.com | 480789.13 | 01009700000 | May | 2023
3333344444 | Mootassem Elzarka | mootassem@gmail.com | 591427.00 | 01234567899 | May | 2019
=====
```

```

>>> 10
Could you please specify the sorting order you prefer for the data to be printed?
1 for sorted by name / 2 for sorted by balance / 3 for sorted by open date
>>> 3
Do you want to print it in table format? yes:1 no:0 0
Sorted by Date (old-new)
Account Number: 9700000003
Name: Micheal Robert
E-mail: michael@yahoo.com
Balance: 300.00 $
Mobile: 01009700003
Opened: November 2008

Account Number: 1111122222
Name: Raphy Takla
E-mail: raphy@gmail.com
Balance: 23771.46 $
Mobile: 01278967897
Opened: September 2011

Account Number: 2222233333
Name: Omar Hekal
E-mail: omar@gmail.com
Balance: 40000.73 $
Mobile: 01122553164
Opened: December 2015

Account Number: 3333344444
Name: Mootassem Elzarka
E-mail: mootassem@gmail.com
Balance: 591427.00 $
Mobile: 01234567899
Opened: May 2019

Account Number: 4444455555
Name: Micheal Jones
E-mail: m.jones@gmail.com
Balance: 480789.13 $
Mobile: 01009700000
Opened: May 2023

Account Number: 9700000102
Name: Abdelrahman Essam
E-mail: abdelrahman@yahoo.com
Balance: 0.00 $
Mobile: 01227894265
Opened: December 2023
=====

```

## XI. Quit (*Press 11*):

- Exit the program.

```

>>> 11

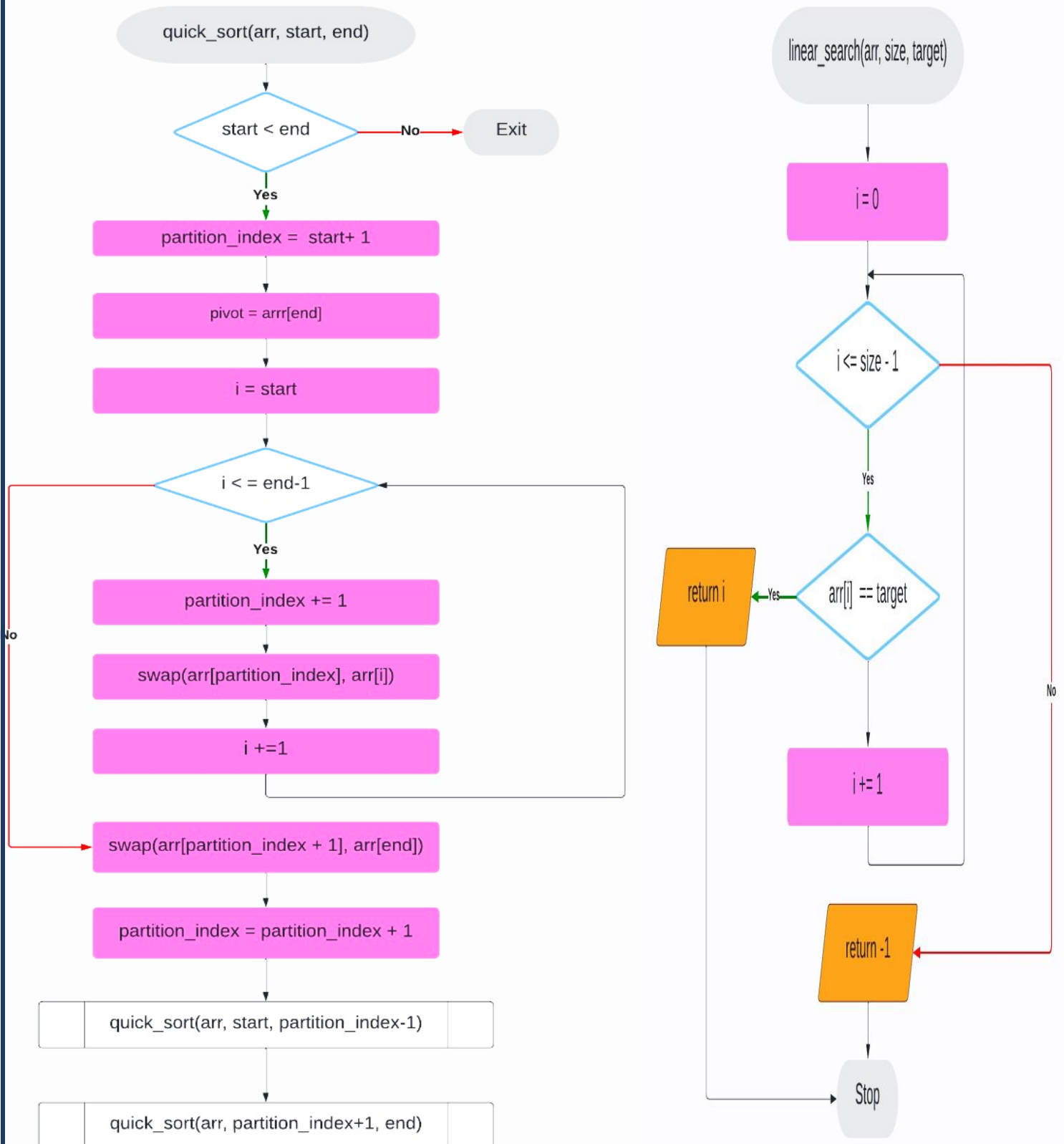
Thank you for using our program.

Process returned 0 (0x0)    execution time : 6.154 s
Press any key to continue.

```

## Main Algorithms used in search and sort

### Flowchart:



## **Pseudocode:**

- Pseudocode for Quick Sort

```
procedure quick_sort(arr, start, end)
    if start < end then
        partition_index := start - 1
        pivot := arr[end] // pick the last element as the pivot

        // Putting bigger elements to the left and smaller elements to the right
        for i := start to end - 1 do
            if arr[i] < pivot then
                partition_index := partition_index + 1
                swap(arr[partition_index], arr[i])

        // Swap the pivot to its correct position
        swap(arr[partition_index + 1], arr[end])

        // Recursive calls for the left and right partitions
        partition_index := partition_index + 1
        quick_sort(arr, start, partition_index - 1) // Sort the left part
        quick_sort(arr, partition_index + 1, end)   // Sort the right part
    end if
end procedure
```

- Pseudocode for Quick Sort

```
procedure linear_search(arr, size, target)
    for i := 0 to size - 1 do
        if arr[i] = target then
            return i // Target found at index i
    return -1 // Target not found in the array
end procedure
```