```
#include <stdio.h>
#include <string.h>

// Define a structure to represent a bank account
struct BankAccount {
    int account_number;
    char name[50];
    float balance;
};

// Function prototypes
void createAccount(struct BankAccount[], int*);
void deposit(struct BankAccount[], int);
void withdraw(struct BankAccount[], int);
void checkBalance(struct BankAccount[], int);
void displayAccount(struct BankAccount[], int);

int main() {
    struct BankAccount accounts[100]; // Array to store bank accounts
    int numAccounts = 0; // Current number of bank accounts

    int choice;

    do {
        // Display menu
        printf("\nMenu:\n");
        printf("1. Create Account\n");
        printf("2. Deposit\n");
        printf("3. Withdraw\n");
        printf("4. Check Balance\n");
```

```
    printf("5. Display Account\n");
    printf("6. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch(choice) {
        case 1:
            createAccount(accounts, &numAccounts);
            break;
        case 2:
            deposit(accounts, numAccounts);
            break;
        case 3:
            withdraw(accounts, numAccounts);
            break;
        case 4:
            checkBalance(accounts, numAccounts);
            break;
        case 5:
            displayAccount(accounts, numAccounts);
            break;
        case 6:
            printf("Exiting program.\n");
            break;
        default:
            printf("Invalid choice. Please try again.\n");
    }
} while(choice != 6);

return 0;
```

```
}

// Function to create a new bank account
void createAccount(struct BankAccount accounts[], int *numAccounts) {
    if (*numAccounts < 100) {
        struct BankAccount newAccount;
        printf("Enter Account Number: ");
        scanf("%d", &newAccount.account_number);

        printf("Enter Name: ");
        scanf(" %[^\n]s", newAccount.name); // Read with spaces

        newAccount.balance = 0;

        accounts[*numAccounts] = newAccount;
        (*numAccounts)++;
        printf("Account created successfully.\n");
    } else {
        printf("Cannot create more accounts. Limit reached.\n");
    }
}

// Function to deposit money into an account
void deposit(struct BankAccount accounts[], int numAccounts) {
    int accountNumber;
    float amount;
    printf("Enter Account Number: ");
    scanf("%d", &accountNumber);

    int found = 0;
```

```
    for (int i = 0; i < numAccounts; i++) {
        if (accounts[i].account_number == accountNumber) {
            printf("Enter Amount to Deposit: ");
            scanf("%f", &amount);
            accounts[i].balance += amount;
            printf("Deposit successful. Current balance: %.2f\n", accounts[i].balance);
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Account with Account Number %d not found.\n", accountNumber);
    }
}

// Function to withdraw money from an account
void withdraw(struct BankAccount accounts[], int numAccounts) {
    int accountNumber;
    float amount;
    printf("Enter Account Number: ");
    scanf("%d", &accountNumber);

    int found = 0;
    for (int i = 0; i < numAccounts; i++) {
        if (accounts[i].account_number == accountNumber) {
            printf("Enter Amount to Withdraw: ");
            scanf("%f", &amount);
            if (amount > accounts[i].balance) {
                printf("Insufficient balance.\n");
            } else {
```

```
            accounts[i].balance -= amount;
            printf("Withdrawal successful. Current balance: %.2f\n", accounts[i].balance);
        }
        found = 1;
        break;
    }
  }
  if (!found) {
    printf("Account with Account Number %d not found.\n", accountNumber);
  }
}

// Function to check balance of an account
void checkBalance(struct BankAccount accounts[], int numAccounts) {
  int accountNumber;
  printf("Enter Account Number: ");
  scanf("%d", &accountNumber);

  int found = 0;
  for (int i = 0; i < numAccounts; i++) {
    if (accounts[i].account_number == accountNumber) {
      printf("Current Balance: %.2f\n", accounts[i].balance);
      found = 1;
      break;
    }
  }
  if (!found) {
    printf("Account with Account Number %d not found.\n", accountNumber);
  }
}
```

```
// Function to display account information
void displayAccount(struct BankAccount accounts[], int numAccounts) {
    int accountNumber;
    printf("Enter Account Number: ");
    scanf("%d", &accountNumber);

    int found = 0;
    for (int i = 0; i < numAccounts; i++) {
        if (accounts[i].account_number == accountNumber) {
            printf("Account Number: %d\n", accounts[i].account_number);
            printf("Name: %s\n", accounts[i].name);
            printf("Balance: %.2f\n", accounts[i].balance);
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Account with Account Number %d not found.\n", accountNumber);
    }
}
```

Output:

```
Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 1
Enter Account Number: 19
Enter Name: Abdul Hadee
Account created successfully.

Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 1
Enter Account Number: 11
Enter Name: Hadee
Account created successfully.
```

```
Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 5
Enter Account Number: 19
Account Number: 19
Name: Abdul Hadee
Balance: 0.00

Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 5
Enter Account Number: 11
Account Number: 11
Name: Hadee
Balance: 0.00
```

```
Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 2
Enter Account Number: 19
Enter Amount to Deposit: 1000000
Deposit successful. Current balance: 1000000.00

Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 3
Enter Account Number: 19
Enter Amount to Withdraw: 9450
Withdrawal successful. Current balance: 990550.00
```

```
Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 2
Enter Account Number: 11
Enter Amount to Deposit: 45000
Deposit successful. Current balance: 45000.00

Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 3
Enter Account Number: 11
Enter Amount to Withdraw: 1400
Withdrawal successful. Current balance: 43600.00
```

```
Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 4
Enter Account Number: 19
Current Balance: 990550.00

Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 4
Enter Account Number: 11
Current Balance: 43600.00
```

```
Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 5
Enter Account Number: 19
Account Number: 19
Name: Abdul Hadee
Balance: 990550.00

Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 5
Enter Account Number: 11
Account Number: 11
Name: Hadee
Balance: 43600.00
```

```
Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Display Account
6. Exit
Enter your choice: 6
Exiting program.
```