```
class  LinearProbing:
    def __init__(self, size):
        self.size = size
        self.keys = [None] * size
        self.collisions = [0] * size

    def hash_func(self, key):
        return key % self.size

    def insert_lp(self, key):
        i = self.hash_func(key)
        i = i

        while self.keys[i] is not None:
            i = (i + 1) % self.size
            self.collisions[i] += 1
            if i == i+1:
                print("Hash table is full. Cannot insert.")
                return

        self.keys[i] = key

    def search_lp(self, key):
        i = self.hash_func(key)
        i = i

        while self.keys[i] is not None:
            if self.keys[i] == key:
                return i
            i = (i + 1) % self.size
            if i == i:
                break
```

```
        return None

    def display(self):
        print("| Index | Key | Collisions |")
        print("|-------|-----|------------|")
        for i in range(self.size):
            print(f"|   {i}   |  {self.keys[i]}  |     {self.collisions[i]}     |")




def main():
    while True:
        print("\nMain Menu")
        print("Select one of these options:")
        print("1) Linear Probing")
        print("2) Exit")

        menu_choice = int(input("Enter your choice: "))

        if menu_choice == 1:
            size = int(input("Enter the size of the hash table: "))
:wq
            submenu(linear_table, "Linear Probing")

        elif menu_choice == 2:
            break
        else:
            print("Invalid choice. Please enter a valid option.")
```

```
def submenu(hash_table, technique):
    while True:
        print(f"\n{technique} Menu")
        print("Select one of these options:")
        print("1) Insert")
        print("2) Search")
        print("3) Display")
        print("4) Return to Main Menu")

        choice = int(input("Enter your choice: "))
:wq
:wq:wq
::Wq
        if choice == 1:
            value = int(input("Enter the value to insert: "))
            hash_table.insert_lp(value)
        elif choice == 2:
            value = int(input("Enter the value to search: "))
            result = hash_table.search_lp(value)
            if result is not None:
                print(f"{technique}: Value found at i {result}.")
            else:
                print(f"{technique}: Value not found.")
        elif choice == 3:
            hash_table.display()
        elif choice == 4:
            break
        else:
            print("Invalid choice. Please enter a valid option.")



if __name__ == "__main__":
```

main():wq