```
class Waypoint:
    def __init__(self, location, description):
        self.location = location
        self.description = description
        self.next = None
        self.prev = None

class Route:
    def __init__(self):
        self.head = None

    def add_waypoint(self, location, description):
        new_waypoint = Waypoint(location, description)
        if not self.head:
            self.head = new_waypoint
        else:
            current = self.head
            while current.next:
                current = current.next
            current.next = new_waypoint

    def insert_waypoint_after(self, target, location, description):
        new_waypoint = Waypoint(location, description)
        current = self.head
        while current:
            if current.location == target.location:
                new_waypoint.next = current.next
                current.next = new_waypoint
                new_waypoint.prev = current
                if new_waypoint.next:
                    new_waypoint.next.prev = new_waypoint
                break
            current = current.next

    def remove_waypoint(self, location):
        current = self.head
        prev = None
        while current:
            if current.location == location:
                if prev:
                    prev.next = current.next
                    if current.next:
                        current.next.prev = prev
```

```
                else:
                    self.head = current.next
                    if self.head:
                        self.head.prev = None
                break
            prev = current
            current = current.next

    def next_waypoint(self):
        if self.head:
            self.head = self.head.next
            return self.head

class BidirectionalRoute(Route):
    def __init__(self):
        super().__init__()

    def previous_waypoint(self):
        if self.head and self.head.prev:
            self.head = self.head.prev
            return self.head

def print_menu():
    print("\nMenu:")
    print("1. Add waypoint")
    print("2. Insert waypoint after")
    print("3. Remove waypoint")
    print("4. Traverse in single direction")
    print("5. Bidirectional traversal")
    print("6. Exit")

def display_route(route):
    print("\nCurrent Route:")
    current = route.head
    while current:
        print(f"Location: {current.location}, Description:
{current.description}")
        current = current.next

# Demonstration
waypoint1 = Waypoint("Location 1", "Description 1")
waypoint2 = Waypoint("Location 2", "Description 2")
waypoint3 = Waypoint("Location 3", "Description 3")
```

```
waypoint4 = Waypoint("Location 4", "Description 4")
waypoint5 = Waypoint("Location 5", "Description 5")

route = Route()
route.add_waypoint("Location 1", "Description 1")
route.add_waypoint("Location 2", "Description 2")
route.add_waypoint("Location 3", "Description 3")
route.add_waypoint("Location 4", "Description 4")
route.add_waypoint("Location 5", "Description 5")

while True:
    print_menu()
    choice = input("Enter your choice: ")

    if choice == "1":
        location = input("Enter location: ")
        description = input("Enter description: ")
        route.add_waypoint(location, description)
        display_route(route)
    elif choice == "2":
        target_location = input("Enter target location: ")
        location = input("Enter location: ")
        description = input("Enter description: ")
        target = None
        current = route.head
        while current:
            if current.location == target_location:
                target = current
                break
            current = current.next
        if target:
            route.insert_waypoint_after(target, location, description)
            display_route(route)
        else:
            print("Target location not found.")
    elif choice == "3":
        location = input("Enter location to remove: ")
        route.remove_waypoint(location)
        display_route(route)
    elif choice == "4":
        print("\nTraversal in single direction:")
        current = route.head
        while current:
```

```
            print(f"Location: {current.location}, Description:
{current.description}")
            current = route.next_waypoint()
    elif choice == "5":
        print("\nBidirectional traversal:")
        bidirectional_route = BidirectionalRoute()
        bidirectional_route.head = route.head
        current = bidirectional_route.head
        while current:
            print(f"Location: {current.location}, Description:
{current.description}")
            current = bidirectional_route.next_waypoint()
        print("\nReversing:")
        current = bidirectional_route.head
        while current:
            print(f"Location: {current.location}, Description:
{current.description}")
            current = bidirectional_route.previous_waypoint()
    elif choice == "6":
        print("Exiting...")
        display_route(route)
        break
    else:
        print("Invalid choice. Please try again.")
```