

**Abbottabad university of science & technology**

**Parking lot management:**

**Submitted by : Fatima bibi**

**Submitted to : Sir Jamal Abdul Ahad**

**Roll No : 14646**

**Course : Data structure & Algorithm**

***PROJECT CODE:***

import heapq # Import heapq for heap operations

# Global Variables

Vehicle\_Number = []

Vehicle\_Type = []

Vehicle\_Name = []

Owner\_Name = []

Date = []

Time\_ = []

# Heaps for parking spaces

bicycle\_spaces = list(range(1, 79)) # 78 spaces

bike\_spaces = list(range(1, 101)) # 100 spaces

car\_spaces = list(range(1, 251)) # 250 spaces

# Convert to heaps

heapq.heapify(bicycle\_spaces)

heapq.heapify(bike\_spaces)

heapq.heapify(car\_spaces)

def get\_vehicle\_number():

"""Get and validate the vehicle number."""

while True:

Vno = input("\tEnter vehicle number (XXXX-XX-XXXX): ").upper()

if len(Vno) == 12 and Vno not in Vehicle\_Number:

return Vno

elif Vno in Vehicle\_Number:

print("###### Vehicle Number Already Exists ######")

else:

print("###### Enter a Valid Vehicle Number ######")

def get\_vehicle\_type():

"""Get and validate the vehicle type."""

while True:

Vtype = input("\tEnter vehicle type (Bicycle=A/Bike=B/Car=C): ").lower()

if Vtype == "a" and bicycle\_spaces:

space = heapq.heappop(bicycle\_spaces) # Allocate the nearest space

return "Bicycle", space

elif Vtype == "b" and bike\_spaces:

space = heapq.heappop(bike\_spaces) # Allocate the nearest space

return "Bike", space

elif Vtype == "c" and car\_spaces:

space = heapq.heappop(car\_spaces) # Allocate the nearest space

return "Car", space

print("###### Invalid or No Space Available ######")

def get\_non\_empty\_input(prompt):

"""Get a non-empty input."""

while True:

value = input(prompt)

if value.strip():

return value

print("###### Input Cannot Be Empty ######")

def get\_date():

"""Get and validate the date in DD-MM-YYYY format."""

while True:

date = input("\tEnter Date (DD-MM-YYYY): ")

if len(date) == 10:

return date

print("###### Enter a Valid Date ######")

def get\_time():

"""Get and validate the time in HH:MM:SS format."""

while True:

time\_ = input("\tEnter Time (HH:MM:SS): ")

if len(time\_) == 8:

return time\_

print("###### Enter a Valid Time ######")

def add\_vehicle():

"""Add a new vehicle entry."""

Vehicle\_Number.append(get\_vehicle\_number())

vtype, space = get\_vehicle\_type()

Vehicle\_Type.append(vtype)

Vehicle\_Name.append(get\_non\_empty\_input("\tEnter vehicle name: "))

Owner\_Name.append(get\_non\_empty\_input("\tEnter owner name: "))

Date.append(get\_date())

Time\_.append(get\_time())

print(f"\n###### {vtype} parked at space {space}. Entry Saved ######")

def remove\_vehicle():

"""Remove a vehicle entry."""

Vno = get\_non\_empty\_input("\tEnter vehicle number to remove: ").upper()

if Vno in Vehicle\_Number:

index = Vehicle\_Number.index(Vno)

vtype = Vehicle\_Type[index]

# Return the space to the appropriate heap

if vtype == "Bicycle":

heapq.heappush(bicycle\_spaces, index + 1)

elif vtype == "Bike":

heapq.heappush(bike\_spaces, index + 1)

elif vtype == "Car":

heapq.heappush(car\_spaces, index + 1)

# Remove the vehicle

Vehicle\_Number.pop(index)

Vehicle\_Type.pop(index)

Vehicle\_Name.pop(index)

Owner\_Name.pop(index)

Date.pop(index)

Time\_.pop(index)

print("\n###### Vehicle Entry Removed ######")

else:

print("###### Vehicle Not Found ######")

def view\_parked\_vehicles():

"""Display all parked vehicles."""

print("Vehicle No.\tType\tName\tOwner\tDate\tTime")

for i in range(len(Vehicle\_Number)):

print(f"{Vehicle\_Number[i]}\t{Vehicle\_Type[i]}\t{Vehicle\_Name[i]}\t{Owner\_Name[i]}\t{Date[i]}\t{Time\_[i]}")

print("###### Total Parked Vehicles:", len(Vehicle\_Number))

def view\_spaces():

"""Display available parking spaces."""

print("\nAvailable Spaces:")

print(f"Bicycles: {len(bicycle\_spaces)}")

print(f"Bikes: {len(bike\_spaces)}")

print(f"Cars: {len(car\_spaces)}")

def main():

"""Main menu loop."""

while True:

print("\n1. Add Vehicle")

print("2. Remove Vehicle")

print("3. View Parked Vehicles")

print("4. View Spaces")

print("5. Exit")

try:

choice = int(input("Choose an option: "))

if choice == 1:

add\_vehicle()

elif choice == 2:

remove\_vehicle()

elif choice == 3:

view\_parked\_vehicles()

elif choice == 4:

view\_spaces()

elif choice == 5:

print("Exiting... Goodbye!")

break

else:

print("Invalid Option!")

except ValueError:

print("Please enter a valid number.")

if \_\_name\_\_ == "\_\_main\_\_":

main()







