Q1. Write a program to create a dictionary containing names of competition winner students as keys and number of their wins as values.

In [1]:
```
def create_winner_dict():
    winner_dict = {}
    num_students = int(input("Enter the number of students: "))

    for i in range(num_students):
        name = input(f"Enter the name of student {i+1}: ")
        wins = int(input(f"Enter the number of wins for {name}: "))
        winner_dict[name] = wins

    return winner_dict

if __name__ == "__main__":
    print('21052410')
    winners = create_winner_dict()
    print("Competition winners dictionary:")
    print(winners)
```

```
21052410
Enter the number of students: 3
Enter the name of student 1: Brijit Adak
Enter the number of wins for Brijit Adak: 4
Enter the name of student 2: Sayan Maity
Enter the number of wins for Sayan Maity: 5
Enter the name of student 3: Diptendra Maity
Enter the number of wins for Diptendra Maity : 6
Competition winners dictionary:
{'Brijit Adak': 4, 'Sayan Maity': 5, 'Diptendra Maity ': 6}
```

Q2. Write a program to create a phone directory for all your friends and then print it.

In [2]:
```
def create_phone_directory():
    phone_directory = {}
    num_friends = int(input("Enter the number of friends: "))

    for i in range(num_friends):
        name = input(f"Enter the name of friend {i+1}: ")
        phone_number = input(f"Enter the phone number for {name}: ")
        phone_directory[name] = phone_number

    return phone_directory

def print_phone_directory(directory):
    print("Phone Directory:")
    for name, phone_number in directory.items():
        print(f"{name}: {phone_number}")

if __name__ == "__main__":
    print('21052410')
    friends_directory = create_phone_directory()
    print_phone_directory(friends_directory)
```

```
21052410
Enter the number of friends: 3
Enter the name of friend 1: Brijit Adak
Enter the phone number for Brijit Adak: 123
Enter the name of friend 2: Sayan Maity
Enter the phone number for Sayan Maity: 234
Enter the name of friend 3: Diptendra Maity
Enter the phone number for Diptendra Maity: 345
Phone Directory:
Brijit Adak: 123
Sayan Maity: 234
Diptendra Maity: 345
```

Q3. Marks of three students ‘Suniti’, ‘Ryna’, and ‘ziva’ in 3 subjects are available in dictionary.
D1 = {1:40, 2:70, 3:70}
D2 = {1:40, 2:50, 3:60}
D3 = {1:70, 2:80, 3:90}
Sample Output:
Key Value
Ryna {1:40, 2:50, 3:60}

Subject(Key)	Marks(Value)
1	40
2	50
3	60

Find the highest mark in subject 3.

In [17]:
```
D1 = {1:40, 2:70, 3:70}
D2 = {1:40, 2:50, 3:60}
D3 = {1:70, 2:80, 3:90}
subject_3_marks = [D1[3], D2[3], D3[3]]
highest_mark = max(subject_3_marks)
print('21052410')
print('Key\tValue')
print('Ryna',D2,sep="\t")
print('Subject(Key)\tMarks(Value)')
for i in D2.items():
    print(i[0],'\t\t',i[1])

print("Highest mark in subject 3:", highest_mark)
```

```
21052410
Key     Value
Ryna    {1: 40, 2: 50, 3: 60}
Subject(Key)    Marks(Value)
1                40
2                50
3                60
Highest mark in subject 3: 90
```

Q4. Create a dictionary whose keys are month names and whose values are the number of days
in the corresponding months.
a) Ask the user to enter a month name and use the dictionary to tell them how many days are in the month.
b) Print out all the keys in alphabetical order.
c) Print out all of the month with 31 days.
d) Print out the (key value) pairs started by the number of days in each month.

In [22]:

```python
def create_month_dictionary():
    months_dict = {
        "january": 31,
        "february": 28,
        "march": 31,
        "april": 30,
        "may": 31,
        "june": 30,
        "july": 31,
        "august": 31,
        "september": 30,
        "actober": 31,
        "november": 30,
        "december": 31
    }
    return months_dict

def main():
    print('21052410')
    months_dict = create_month_dictionary()

    # a)
    month_name = input("Enter a month name: ").lower()
    if month_name in months_dict:
        print(f"{month_name} has {months_dict[month_name]} days.")
    else:
        print("Invalid month name.")

    # b)
    print("\nMonth names in alphabetical order:")
    for month in sorted(months_dict.keys()):
        print(month)

    # c)
    print("\nMonths with 31 days:")
    for month, days in months_dict.items():
        if days == 31:
            print(month)

    # d)
    print("\n(Key, Value) pairs sorted by the number of days:")
    sorted_pairs = sorted(months_dict.items(), key=lambda x: x[1])
    for month, days in sorted_pairs:
        print(f"{month}: {days}")

if __name__ == "__main__":
    main()
```

```
21052410
Enter a month name: July
july has 31 days.

Month names in alphabetical order:
actober
april
august
december
february
january
july
june
march
may
november
september

Months with 31 days:
january
march
may
july
august
actober
december

(Key, Value) pairs sorted by the number of days:
february: 28
april: 30
june: 30
september: 30
november: 30
january: 31
march: 31
may: 31
july: 31
august: 31
actober: 31
december: 31
```

Q5. Repeatedly ask the user to enter a team name and how many games the team has won and how many they lost.
Store this information in a dictionary where the keys are the team names and the values are list
of the form [wins, losses]
a) Using the dictionary created above, allow the user to enter a team name and print out the team's winning percentage.
b) Using the dictionary, create a list whose entries are the member of wins of each team.
c) Using the dictionary, create a list of all those teams that have winning records.

In [28]:

```
def create_team_dictionary():
    team_dict = {}
    while True:
        team_name = input("Enter the team name (or type 'done' to finis
        if team_name.lower() == 'done':
            break
        wins = int(input("Enter the number of wins for the team: "))
        losses = int(input("Enter the number of losses for the team: ")
        team_dict[team_name] = [wins, losses]
    return team_dict

def calculate_winning_percentage(team_dict, team_name):
    if team_name in team_dict:
        wins, losses = team_dict[team_name]
        total_games = wins + losses
        if total_games > 0:
            winning_percentage = (wins / total_games) * 100
            return winning_percentage
        else:
            return 0
    else:
        return None

def main():

    print('21052410')
    team_dict = create_team_dictionary()

    # a)
    team_name = input("\nEnter a team name to get its winning percentag
    winning_percentage = calculate_winning_percentage(team_dict, team_n
    if winning_percentage is not None:
        print(f"The winning percentage of {team_name} is: {winning_perc
    else:
        print("Team not found.")

    # b)
    wins_list = [wins for wins, _ in team_dict.values()]
    print("\nList of wins for each team:", wins_list)

    # c)
    winning_teams = [team for team, record in team_dict.items() if recc
    print("\nTeams with winning records:", winning_teams)

if __name__ == "__main__":
    main()
```

```
21052410
Enter the team name (or type 'done' to finish): a
Enter the number of wins for the team: 3
Enter the number of losses for the team: 2
Enter the team name (or type 'done' to finish): b
Enter the number of wins for the team: 2
Enter the number of losses for the team: 3
Enter the team name (or type 'done' to finish): c
Enter the number of wins for the team: 4
Enter the number of losses for the team: 1
Enter the team name (or type 'done' to finish): done

Enter a team name to get its winning percentage: b
The winning percentage of b is: 40.00%

List of wins for each team: [3, 2, 4]

Teams with winning records: ['a', 'c']
```