

Question 1

Write a program which computes the number of game points collected by a rugby team during a match. In rugby, you can score game points in four different ways.

- Each try is awarded 5 points
- Each conversion is awarded 2 points
- Each penalty is awarded 3 points
- Each drop goal is awarded 3 points

A conversion can only be attempted if a try is scored.

Your program should input four numbers (the number of tries, conversions, penalties and drop goals). It should then print the total number of game points collected, by multiplying the input numbers by the appropriate points. If the number of conversions is greater than the number of tries, the input is invalid. In such a case the program should print an error message to the user.

Example 1:

Number of tries? 3
Number of conversions? 2
Number of penalties? 4
Number of drop goals? 0
Game points: 31

Example 2:

Number of tries? 2
Number of conversions? 3
Impossible: too many conversions or too few tries.

```
def main():
    t = int(input("number of tries?"))
    c = int(input("number of conversions?"))
    if c > t:
        print("Error!")
    else:
        p = int(input("number of penalties?"))
        d = int(input("number of drop goals?"))
        print("Total score", t*5+c*2+3*(p+d))

main()
```

Question 2

A tournament matrix T contains all the scores of the matches played between each pair of teams in a tournament. Element $T[i][j]$ contains the score difference of team i against team j . (This means that $T[i][j] = -T[j][i]$.)

Read a file containing the scores for each match, and construct a tournament matrix. In this file, the teams are identified by number (not by name) Each line contains the numbers of the two teams, followed by the corresponding scores.

Example input file ("example")

```
1 2 23 12
3 4 10 34
1 3 67 0
2 4 21 7
1 4 6 0
2 3 15 7
```

Implement a function `tournament(n, file_name)` which reads the data from the file with the given name and returns a tournament matrix as a list of lists. The parameter `n` is the number of teams.

Hints:

- python lists are indexed from zero, so don't forget to subtract one to the team number when accessing the matrix.
- Don't forget that the diagonal should be filled with zeros.

Solution and output on the example:

```
def tournament(n, fname):
    t = []
    for i in range(n):
        t.append([0]*n)
    # print(t)
    with open(fname) as f:
        for line in f:
            i,j,x,y = [int(x) for x in line.split()]
            i = i-1
            j = j-1
            t[i][j] = x - y
            t[j][i] = y - x
    return t

print(tournament(4,"example"))
```

```
[[0, 11, 67, 6], [-11, 0, 8, 14], [-67, -8, 0, -24], [-6, -14, 24, 0]]
```

Question 3

A rugby team is composed of 23 players, but only 15 players are on the pitch at a given time. An interesting statistics is the number of minutes played by each player in a given match.

Define a class `Team` with the following methods:

- `__init__(self, pitch_players, replacement_players)` which initializes the team object. The argument specifies which players are on the pitch at the start of the game, and which are replacement.
- `swap(self, player_1, player_2)` which removes player 1 from the pitch and puts player 2 from the bench in their place.
- `tick(self)` which is assumed to be executed once every minute. It increases by one the number of minutes played for each player on the pitch.
- `print_stats(self)` which prints the play time of every player in the team.

Here comes an example of use of the class and the corresponding output:

```
t = Team(["Alana", "Chelsea", "Georgia"], ["Iritana", "Kate"])
t.tick()
t.tick()
t.tick()
t.swap("Alana", "Iritana")
t.tick()
t.tick()
t.tick()
t.tick()
t.print_stats()
```

Alana played 3 minutes Chelsea played 7 minutes Georgia played 7 minutes
Iritana played 4 minutes Kate played 0 minutes

Solution and example:

```

class Team:
    def __init__(self, pitch, bench):
        self.pitch = pitch
        self.bench = bench
        self.minutes = dict()
        for p in pitch+bench:
            self.minutes[p] = 0
    def tick(self):
        for p in self.pitch:
            self.minutes[p] += 1
    def swap(self,p1,p2):
        self.pitch.remove(p1)
        self.pitch.append(p2)
        self.bench.remove(p2)
        self.bench.append(p1)
    def print_stats(self):
        for p in self.minutes:
            print(p,"played",self.minutes[p],"minutes")

t = Team(["Alana","Chelsea","Georgia"],["Iritana","Kate"])
t.tick()
t.tick()
t.tick()
t.swap("Alana","Iritana")
t.tick()
t.tick()
t.tick()
t.tick()
t.print_stats()

```

```

Alana played 3 minutes
Chelsea played 7 minutes
Georgia played 7 minutes
Iritana played 4 minutes
Kate played 0 minutes

```