```
import pandas as pd
import numpy as np
```

```
courses = pd.read_csv('/content/courses.csv')
students = pd.read_csv('/content/students.csv')
nov = pd.read_csv('/content/reg-month1.csv')
dec = pd.read_csv('/content/reg-month2.csv')

matches = pd.read_csv('/content/matches.csv')
delivery = pd.read_csv('/content/deliveries.csv')
```

```
dec
```

```
# pd.concat
# df.concat
# ignore_index
# df.append
# mullitindex -> fetch using iloc
# concat dataframes horizontally
```

```
regs = pd.concat([nov,dec],ignore_index=True)
regs
```

```
nov.append(dec,ignore_index=True)
```

```
multi = pd.concat([nov,dec],keys=['Nov','Dec'])
# Multiindex DataFrame
multi.loc[('Dec',4)]
```

```
pd.concat([nov,dec],axis=1)
```

```
# inner join
students.merge(regs,how='inner',on='student_id')
```

```
# left join
courses.merge(regs,how='left',on='course_id')
```

```
# right join
temp_df = pd.DataFrame({
    'student_id':[26,27,28],
    'name':['Nitish','Ankit','Rahul'],
    'partner':[28,26,17]
})

students = pd.concat([students,temp_df],ignore_index=True)
```

```
students.tail()
```

```
students.merge(regs,how='right',on='student_id')
```

```
regs.merge(students,how='left',on='student_id')
```

```
# outer join
students.merge(regs,how='outer',on='student_id').tail(10)
```

```
# 1. find total revenue generated
total = regs.merge(courses,how='inner',on='course_id')['price'].sum()
total
```

```
# 2. find month by month revenue
temp_df = pd.concat([nov,dec],keys=['Nov','Dec']).reset_index()
temp_df.merge(courses,on='course_id').groupby('level_0')['price'].sum()
```

```
# 3. Print the registration table
# cols -> name -> course -> price
regs.merge(students,on='student_id').merge(courses,on='course_id')[['name','course_name','price']]
```

```
# 4. Plot bar chart for revenue/course
regs.merge(courses,on='course_id').groupby('course_name')['price'].sum().plot(kind='bar')
```

```
# 5. find students who enrolled in both the months
common_student_id = np.intersect1d(nov['student_id'],dec['student_id'])
common_student_id
```

```
students[students['student_id'].isin(common_student_id)]
```

```
# 6. find course that got no enrollment
# courses['course_id']
# regs['course_id']

course_id_list = np.setdiff1d(courses['course_id'],regs['course_id'])
courses[courses['course_id'].isin(course_id_list)]
```

```
# 7. find students who did not enroll into any courses
student_id_list = np.setdiff1d(students['student_id'],regs['student_id'])
students[students['student_id'].isin(student_id_list)].shape[0]

(10/28)*100
```

```
students
```

```
# 8. Print student name -> partner name for all enrolled students
# self join
students.merge(students,how='inner',left_on='partner',right_on='student_id')[['name_x','name_y']]
```

```
# 9. find top 3 students who did most number enrollments
regs.merge(students,on='student_id').groupby(['student_id','name'])['name'].count().sort_values(ascending=Fa
# 10. find top 3 students who spent most amount of money on courses
regs.merge(students,on='student_id').merge(courses,on='course_id').groupby(['student_id','name'])['price'].s
```

```
# Alternate syntax for merge
# students.merge(regs)

pd.merge(students,regs,how='inner',on='student_id')
```

```
# IPL Problems

# find top 3 studiums with highest sixes/match ratio
# find orange cap holder of all the seasons
```

```
matches
```

```
delivery
```

```
temp_df = delivery.merge(matches,left_on='match_id',right_on='id')
```

```
six_df = temp_df[temp_df['batsman_runs'] == 6]
```

```
# stadium -> sixes
num_sixes = six_df.groupby('venue')['venue'].count()
```

```
num_matches = matches['venue'].value_counts()
```

```
(num_sixes/num_matches).sort_values(ascending=False).head(10)
```

```
matches
```

```
temp_df.groupby(['season','batsman'])['batsman_runs'].sum().reset_index().sort_values('batsman_runs',ascendi
```

```
temp_df.groupby(['season','batsman'])['batsman_runs'].sum().reset_index().sort_values('batsman_runs',ascendi
```