Bahria University

Karachi Campus



LAB EXPERIMENT NO.

03

LIST OF TASKS

|  |  |
| --- | --- |
| **TASK NO** | **OBJECTIVE** |
|  | Calculate grade of each student |
|  | Calculate status of each student (Pass or Fail). |
|  | Calculate number of students who have passed and failed. |
|  | Calculate average percentage. |
|  | Top three games that are popular among teens. |
|  | Report five most downloaded applications in social media category. |
|  | How many applications have never been rated. |
|  | Report top paid application of each category. |
|  | Which are the top three categories famous among adults. |
|  | Report all applications of "google". |

Submitted On:

**20-10-2023**

(Date: DD/MM/YY)

**Task # 01:**

**Solution:**

**CODE:**

open\_file=open("Downloads/data.csv",encoding="utf-8")

import csv

read\_file=csv.reader(open\_file)

read\_file

dataset = list(read\_file)

Average\_Percentage=0

dataset[0].append("Obtain Marks")

dataset[0].append("Percentage")

dataset[0].append("Remark")

dataset[0].append("Grade")

for i in dataset[1:]:

chem=int(i[dataset[0].index("Chemistry")])

phy=int(i[dataset[0].index("Physics")])

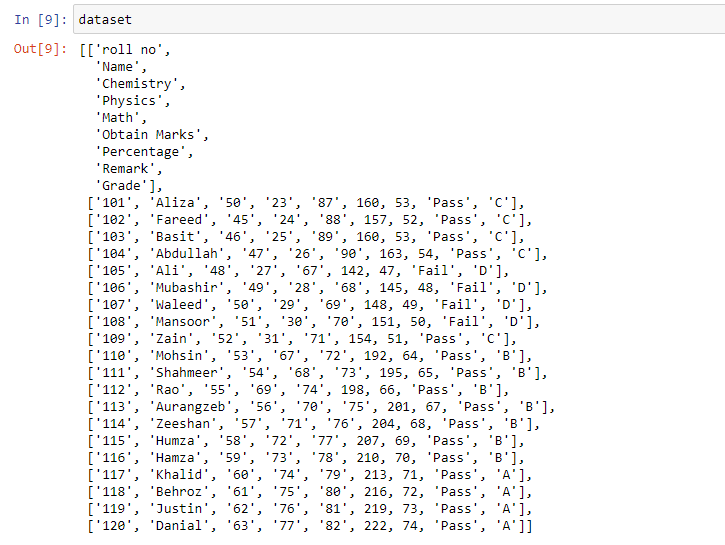
math=int(i[dataset[0].index("Math")])

obtain\_mark=round(chem+phy+math)

percentage=(round(obtain\_mark/300\*100))

Average\_Percentage+=percentage

**OUTPUT:**



**Task#02:**

**Solution:**

**CODE:**

Continued:

if(percentage>80):

Grade="A+"

elif(percentage>70 and percentage<80):

Grade="A"

elif(percentage>60 and percentage<70):

Grade="B"

elif(percentage>50 and percentage<60):

Grade="C"

elif(percentage>40 and percentage<50):

Grade="D"

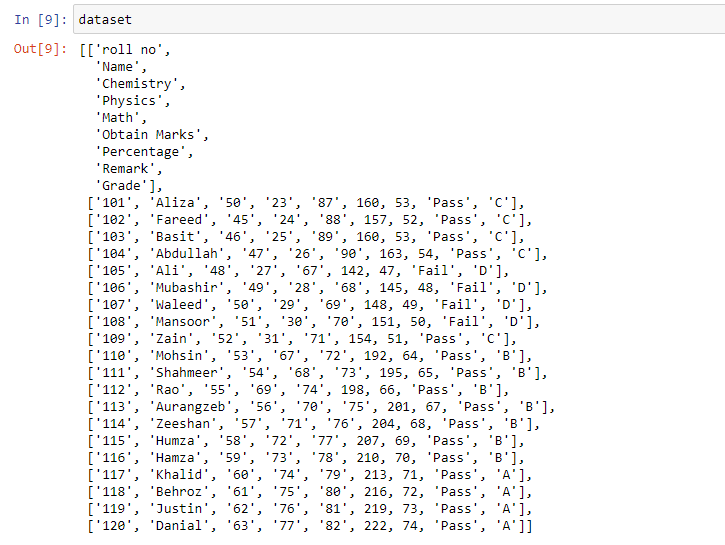
i.append(obtain\_mark)

i.append(percentage)

i.append(remark)

i.append(Grade)

**output:**



**Task # 03:**

**Solution:**

**CODE:**

pass\_count = 0

fail\_count = 0

for row in dataset[1:]:

if row[7] == 'Pass':

pass\_count += 1

else:

fail\_count += 1

Total\_Student\_PASS\_FAIL={

"PASS":pass\_count,

"FAIL":fail\_count,

}

print("Average Percentage ",round(Average\_Percentage/len(dataset)-1,2))

Total\_Student\_PASS\_FAIL

**output:**



**Task#04:**

**Solution:**

**CODE:**

pass\_count = 0

fail\_count = 0

for row in dataset[1:]:

if row[7] == 'Pass':

pass\_count += 1

else:

fail\_count += 1

Total\_Student\_PASS\_FAIL={

"PASS":pass\_count,

"FAIL":fail\_count,

}

print("Average Percentage ",round(Average\_Percentage/len(dataset)-1,2))

Total\_Student\_PASS\_FAIL

**OUTPUT:**

A black text on a white background

Description automatically generated

**Task # 05:**

**Solution:**

**CODE:**

Games=[]

for i in dataset[1:]:

if(i[dataset[0].index("prime\_genre")]=='Games'):

Games.append(i[dataset[0].index("track\_name")])

Games.append(i[dataset[0].index("sup\_devices.num")])

totalgames=[(Games[i],int(Games[i+1])) for i in range(0 ,len(Games),2 )]

totalgames

totalgames.sort(key=lambda x: x[1], reverse=True)

top\_three\_games = totalgames[:3]

for app, downloads in top\_three\_games:

print(f'{app}: {downloads} downloads')

**OUTPUT:**

A black text on a white background

Description automatically generated

**Task # 06:**

**Solution:**

**CODE:**

Most\_Downloaded=[]

for i in dataset[1:]:

if(i[dataset[0].index("prime\_genre")]=='Social Networking'):

Most\_Downloaded.append(i[dataset[0].index("track\_name")])

Most\_Downloaded.append(i[dataset[0].index("sup\_devices.num")])

Most\_Downloaded

app\_downloads = [(Most\_Downloaded[i], int(Most\_Downloaded[i + 1])) for i in range(0, len(Most\_Downloaded), 2)]

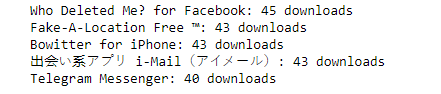
app\_downloads.sort(key=lambda x: x[1], reverse=True)

top\_five\_apps = app\_downloads[:5]

for app, downloads in top\_five\_apps:

print(f'{app}: {downloads} downloads')

**OUTPUT:**



**Task # 07:**

**Solution:**

**CODE:**

Application\_Name\_Not\_Rating=[]

for i in dataset[1:]:

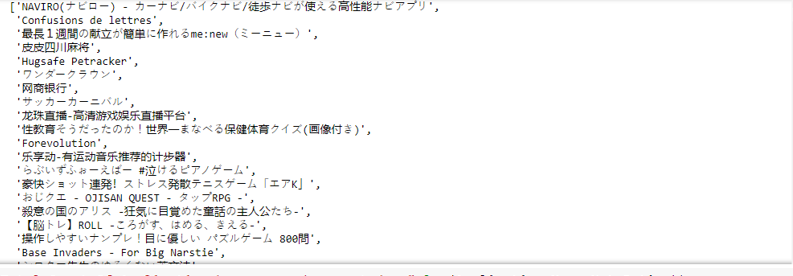
if(float(i[dataset[0].index("user\_rating")])==0):

Application\_Name\_Not\_Rating.append(i[dataset[0].index("track\_name")])

Application\_Name\_Not\_Rating

len(Application\_Name\_Not\_Rating)

**OUTPUT:**

****

****

**Task # 08:**

**Solution:**

**CODE:**

categories = set()

for row in dataset:

categories.add(row[11])

for category in categories:

top\_paid\_app = None

for row in dataset:

if row[11] == category and row[4] != '0':

if top\_paid\_app is None or float(row[4]) > float(top\_paid\_app[4]):

top\_paid\_app = row

if top\_paid\_app is not None:

print(f'Top paid application in {category} category: {top\_paid\_app[1]}')

**OUTPUT:**

A white background with black text

Description automatically generated

**Task # 09:**

**Solution:**

**CODE:**

adult\_categories = []

for row in dataset:

if row[10] == '17+':

adult\_categories.append(row[11])

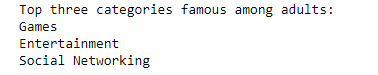
top\_adult\_categories = sorted(set(adult\_categories), key=lambda x: adult\_categories.count(x), reverse=True)[:3]

print('Top three categories famous among adults:')

for category in top\_adult\_categories:

print(category)

**OUTPUT:**



**Task # 10:**

**Solution:**

**CODE:**

AllApplicationofGoogle=[]

for i in dataset[1:]:

if('Google ' in i[dataset[0].index("track\_name")]):

AllApplicationofGoogle.append(i[dataset[0].index("track\_name")])

**OUTPUT:**

