Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
AAD Practical 7

Institute of Computer Technology
B. Tech Computer Science and Engineering

Sub: Algorithm Analysis and Design

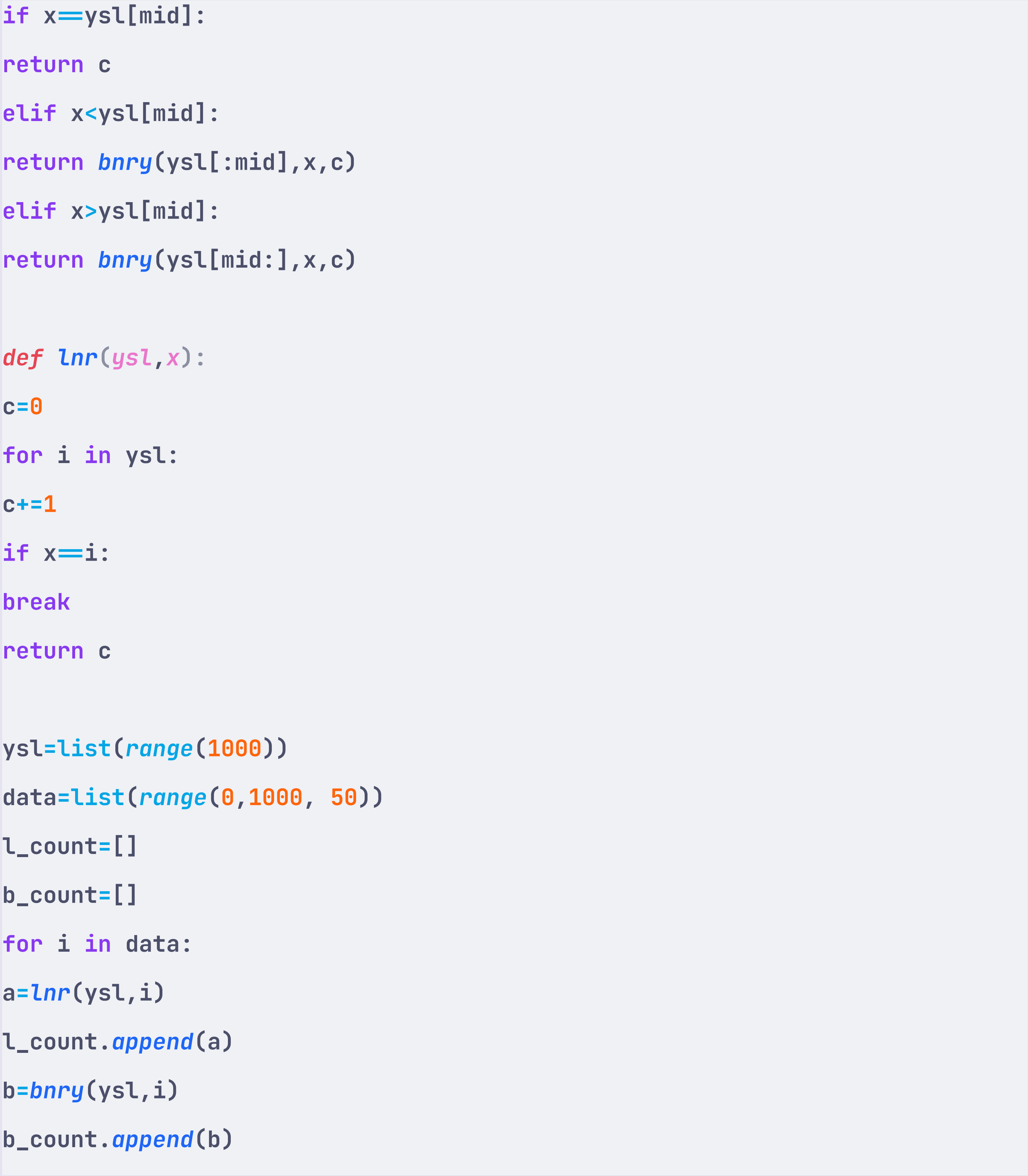
Practical 7

Problem : Trigent is an early pioneer in IT outsourcing and offshore software development business. Thousands of employees working in this company kindly help to find out the employee's details (i.e employee ID, employee salary etc) to implement Recursive Binary search and Linear search (or Sequential Search) and determine the time taken to search an element. Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n. Design the algorithm for the same and implement using the programming language of your choice. Make comparative analysis for various use cases & input size.

Code :

```
import matplotlib

import matplotlib.pyplot as plt

import YSL_io


matplotlib.use('QtAgg')


def bnry(ysl,x,c=0):

c+=1

n=len(ysl)

mid=int(n//2)
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
AAD Practical 7



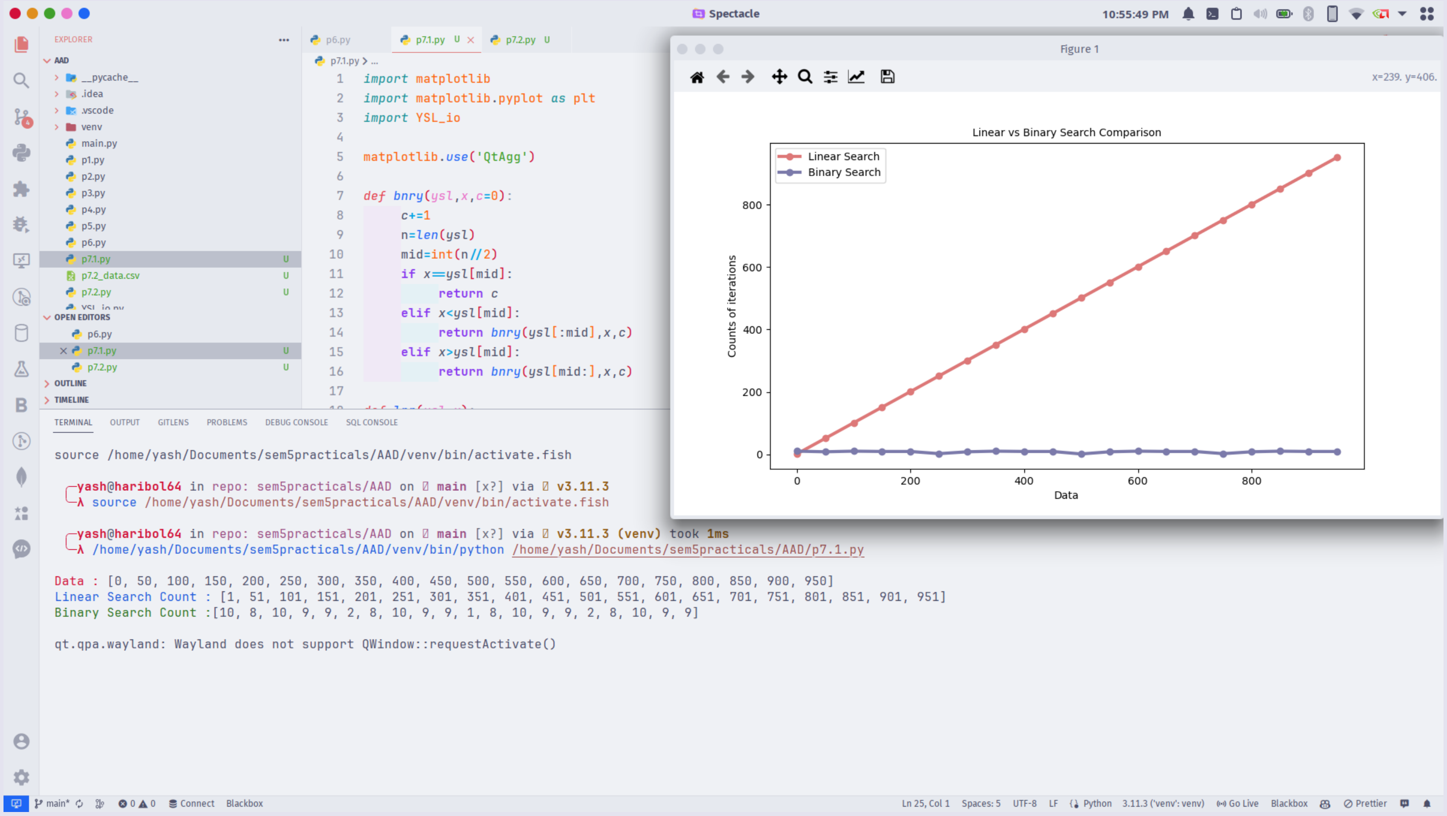
```
if x==ysl[mid]:

return c

elif x<ysl[mid]:

return bnry(ysl[:mid],x,c)

elif x>ysl[mid]:

return bnry(ysl[mid:],x,c)


def lnr(ysl,x):

c=0

for i in ysl:

c+=1

if x==i:

break

return c


ysl=list(range(1000))

data=list(range(0,1000, 50))

l_count=[]

b_count=[]

for i in data:

a=lnr(ysl,i)

l_count.append(a)

b=bnry(ysl,i)

b_count.append(b)
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
AAD Practical 7

```python
YSL_io.printRED("\nData :",end=' ')
print(data)
YSL_io.printBLU("Linear Search Count :",end=' ')
print(l_count)
YSL_io.printGRN(f"Binary Search Count :",end='')
print(b_count)

plt.plot(data,l_count,marker='o',label='Linear Search', color='#dd7878', linewidth=3)
plt.plot(data,b_count,marker='o',label='Binary Search', color='#7b7aaa', linewidth=3)
plt.legend()
plt.title("Linear vs Binary Search Comparison", fontsize=10)
plt.xlabel('Data')
plt.ylabel('Counts of iterations')
print()
plt.show()
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
AAD Practical 7

Screenshot :



```python
import matplotlib
import matplotlib.pyplot as plt
import YSL_io

matplotlib.use('QtAgg')

def bnry(ysl,x,c=0):
    c+=1
    n=len(ysl)
    mid=int(n//2)
    if x==ysl[mid]:
        return c
    elif x<ysl[mid]:
        return bnry(ysl[:mid],x,c)
    elif x>ysl[mid]:
        return bnry(ysl[mid:],x,c)

```

```
source /home/yash/Documents/sem5practicals/AAD/venv/bin/activate.fish

 ┌yash@haribol64 in repo: sem5practicals/AAD on ▯ main [x?] via ▯ v3.11.3
 └λ source /home/yash/Documents/sem5practicals/AAD/venv/bin/activate.fish

 ┌yash@haribol64 in repo: sem5practicals/AAD on ▯ main [x?] via ▯ v3.11.3 (venv) took 1ms
 └λ /home/yash/Documents/sem5practicals/AAD/venv/bin/python /home/yash/Documents/sem5practicals/AAD/p7.1.py

Data : [0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950]
Linear Search Count : [1, 51, 101, 151, 201, 251, 301, 351, 401, 451, 501, 551, 601, 651, 701, 751, 801, 851, 901, 951]
Binary Search Count :[10, 8, 10, 9, 9, 2, 8, 10, 9, 9, 1, 8, 10, 9, 9, 2, 8, 10, 9, 9]

qt.qpa.wayland: Wayland does not support QWindow::requestActivate()
```

Using the algorithm search for the following

1. The designation which has highest salary package
2. The Name of the Employee who has the lowest salary
3. The Mobile number who is youngest employee
4. Salary of the employee who is oldest in age

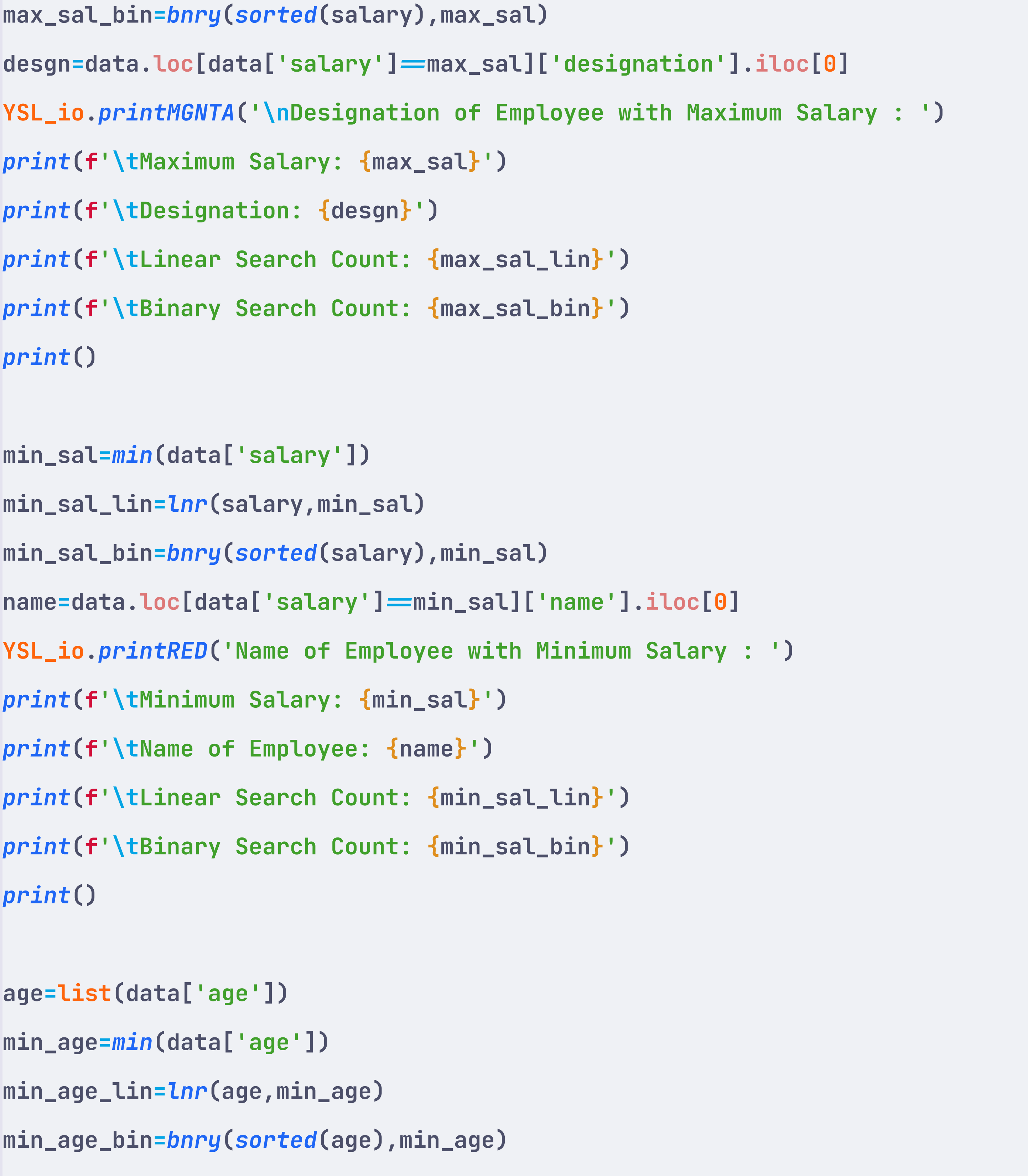
Code :

```python
import pandas as pd
import YSL_io


def bnry(ysl,x,c=0):
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
AAD Practical 7

```
c+=1

n=len(ysl)

mid=int(n//2)

if x==ysl[mid]:

return c

elif x<ysl[mid]:

return bnry(ysl[:mid],x,c)

elif x>ysl[mid]:

return bnry(ysl[mid:],x,c)


def lnr(ysl,x):

c=0

for i in ysl:

c+=1

if x==i:

break

return c

data=pd.read_csv('./p7.2_data.csv')

print()

YSL_io.printGRN(data.set_index('id'))


salary=list(data['salary'])

max_sal=max(data['salary'])

max_sal_lin=lnr(salary,max_sal)
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
AAD Practical 7



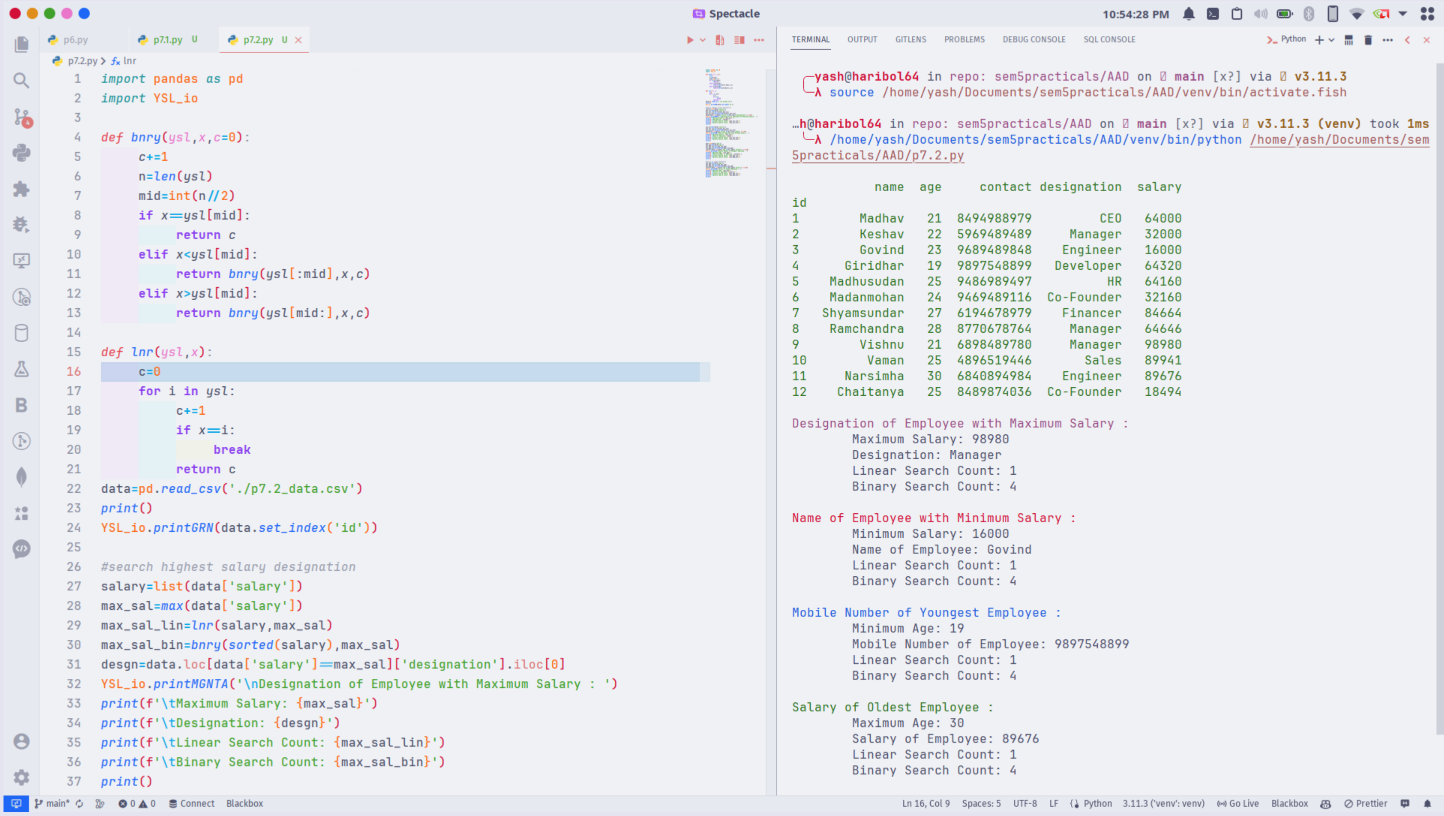
```python
max_sal_bin=bnry(sorted(salary),max_sal)
desgn=data.loc[data['salary']==max_sal]['designation'].iloc[0]
YSL_io.printMGNTA('\nDesignation of Employee with Maximum Salary : ')
print(f'\tMaximum Salary: {max_sal}')
print(f'\tDesignation: {desgn}')
print(f'\tLinear Search Count: {max_sal_lin}')
print(f'\tBinary Search Count: {max_sal_bin}')
print()

min_sal=min(data['salary'])
min_sal_lin=lnr(salary,min_sal)
min_sal_bin=bnry(sorted(salary),min_sal)
name=data.loc[data['salary']==min_sal]['name'].iloc[0]
YSL_io.printRED('Name of Employee with Minimum Salary : ')
print(f'\tMinimum Salary: {min_sal}')
print(f'\tName of Employee: {name}')
print(f'\tLinear Search Count: {min_sal_lin}')
print(f'\tBinary Search Count: {min_sal_bin}')
print()

age=list(data['age'])
min_age=min(data['age'])
min_age_lin=lnr(age,min_age)
min_age_bin=bnry(sorted(age),min_age)
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
AAD Practical 7

```
contact=data.loc[data['age']==min_age]['contact'].iloc[0]

YSL_io.printBLU('Mobile Number of Youngest Employee : ')

print(f'\tMinimum Age: {min_age}')

print(f'\tMobile Number of Employee: {contact}')

print(f'\tLinear Search Count: {min_age_lin}')

print(f'\tBinary Search Count: {min_age_bin}')

print()


max_age=max(data['age'])

max_age_lin=lnr(age,max_age)

max_age_bin=bnry(sorted(age),max_age)

salary1=data.loc[data['age']==max_age]['salary'].iloc[0]

YSL_io.printGRN('Salary of Oldest Employee : ')

print(f'\tMaximum Age: {max_age}')

print(f'\tSalary of Employee: {salary1}')

print(f'\tLinear Search Count: {max_age_lin}')

print(f'\tBinary Search Count: {max_age_bin}')

print()
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
AAD Practical 7

Screenshot :



```python
import pandas as pd
import YSL_io

def bnry(ysl,x,c=0):
     c+=1
     n=len(ysl)
     mid=int(n//2)
     if x==ysl[mid]:
          return c
     elif x<ysl[mid]:
          return bnry(ysl[:mid],x,c)
     elif x>ysl[mid]:
          return bnry(ysl[mid:],x,c)

def lnr(ysl,x):
     c=0
     for i in ysl:
          c+=1
          if x==i:
               break
          return c
data=pd.read_csv('./p7.2_data.csv')
print()
YSL_io.printGRN(data.set_index('id'))

#search highest salary designation
salary=list(data['salary'])
max_sal=max(data['salary'])
max_sal_lin=lnr(salary,max_sal)
max_sal_bin=bnry(sorted(salary),max_sal)
desgn=data.loc[data['salary']==max_sal]['designation'].iloc[0]
YSL_io.printMGNTA('\nDesignation of Employee with Maximum Salary : ')
print(f'\tMaximum Salary: {max_sal}')
print(f'\tDesignation: {desgn}')
print(f'\tLinear Search Count: {max_sal_lin}')
print(f'\tBinary Search Count: {max_sal_bin}')
print()
```

```
╭─yash@haribol64 in repo: sem5practicals/AAD on  main [x?] via  v3.11.3
╰─λ source /home/yash/Documents/sem5practicals/AAD/venv/bin/activate.fish

…h@haribol64 in repo: sem5practicals/AAD on  main [x?] via  v3.11.3 (venv) took 1ms
 ╰─λ /home/yash/Documents/sem5practicals/AAD/venv/bin/python /home/yash/Documents/sem
5practicals/AAD/p7.2.py

          name  age     contact designation  salary
id
1       Madhav   21  8494988979         CEO   64000
2       Keshav   22  5969489489     Manager   32000
3       Govind   23  9689489848    Engineer   16000
4     Giridhar   19  9897548899   Developer   64320
5    Madhusudan  25  9486989497          HR   64160
6    Madanmohan  24  9469489116  Co-Founder   32160
7   Shyamsundar  27  6194678979    Financer   84664
8    Ramchandra  28  8770678764     Manager   64646
9       Vishnu   21  6898489780     Manager   98980
10       Vaman   25  4896519446       Sales   89941
11    Narsimha   30  6840894984    Engineer   89676
12   Chaitanya   25  8489874036  Co-Founder   18494

Designation of Employee with Maximum Salary :
        Maximum Salary: 98980
        Designation: Manager
        Linear Search Count: 1
        Binary Search Count: 4

Name of Employee with Minimum Salary :
        Minimum Salary: 16000
        Name of Employee: Govind
        Linear Search Count: 1
        Binary Search Count: 4

Mobile Number of Youngest Employee :
        Minimum Age: 19
        Mobile Number of Employee: 9897548899
        Linear Search Count: 1
        Binary Search Count: 4

Salary of Oldest Employee :
        Maximum Age: 30
        Salary of Employee: 89676
        Linear Search Count: 1
        Binary Search Count: 4
```