Python (2)

Exit

```python
def fact_rec (n):
  if n==0 or n==1:
    return 1
  else:
    return n*fact_rec (n-1)
number =2
res=fact_rec (number)
print("the factorial of {} is {}".format(number,res))
```

Ln 1, Col 1 History

main.py

Run

Python (3) ⋮ Exit

```python
def isleapyear(year):
    if(year%4==0 and year%100!=0)or year%400==0:
        return True
    else:
        return False
year=int(input("enter a year"))
if isleapyear(year):
  print("{} is leap year".format(year))
else:
  print("{}is not a leap year".format(year))
```

Ln 1, Col 1 History

main.py ⋮

```python
class player:
    def play(self):
        print("The player is playing cricket")

class Batsman(player):
    def play(self):
        print("The batsman is playing batting")

class Blower(player):
    def play(self):
        print("The blower is playing bowling")

batsman = Batsman()
blower = Blower()

batsman.play()
blower.play()
```

Ln 1, Col 1 History

main.py

```
> python3 main.py
The batsman is playing batting
The blower is playing bowling
>
```

Console

Naan mudhalvan4

```python
class BankAccount:
  def __init__(self,account_number, account_holder_name, inital_balance=0.0):
    self.__account_number = account_number
    self.__account_holder_name = account_holder_name
    self.__account_balance = inital_balance

  def deposit(self, amount):
    if amount > 0:
      self.__account_balance += amount
      print("Deposit ₹{}. New balance: {}".format(amount, self.__account_balance))

    else:
      print("Invalid deposit amount")

```

Ln 1, Col 1 • Spaces: 2 History

main.py

Run

Naan mudhalvan4

```python
  def withdraw(self, amount):
    if amount > 0 and amount <= self.__account_balance:
      self.__account_balance -= amount
      print("Withdraw ₹{}. New balance: {}".format(amount, self.__account_balance))

    else:
      print("Invalid withdraw amount")

  def display_balance(self):
    print("Account balance for {} (Account #{}): ₹{}".format(
        self.__account_holder_name, self.__account_number,
        self.__account_balance))
account = BankAccount(account_number="4455769",account_holder_name="Kutdraleshwaran N",
```

Ln 1, Col 1 • Spaces: 2 History

main.py

Run

Naan mudhalvan4

```python
    print("Account balance for {} (Account #{}): ₹{}".format(
        self.__account_holder_name, self.__account_number,
        self.__account_balance))
account = BankAccount(account_number="4455769",account_holder_name="Kutdraleshwaran N",
inital_balance=5000.0)

account.display_balance()
account.deposit(500)
account.withdraw(300)
account.display_balance()
```

Ln 1, Col 1 • Spaces: 2 History

main.py

Run

Naan mudhalvan4

Run 245ms on 17:44:04, 10/18

```
Account balance for Kutdraleshwaran N (
Account #4455769): ₹5000.0
Deposit ₹500. New balance: 5500.0
Withdraw ₹300. New balance: 5200.0
Account balance for Kutdraleshwaran N (
Account #4455769): ₹5200.0
```

Console

Run

Naan mudhalvan

```python
class Student:

  def __init__(self,name,roll_number,cgpa):
    self.name=name
    self.roll_number=roll_number
    self.cgpa=cgpa

def sort_students(student_list):


sort_students=sorted(student_list, key=lambda student: student.cgpa,reverse=True)

    return sort_students

students =[
  Student("mani","A123",7.8),
  Student("ram","A124",9.8),
  Student("vik","A628",8.8),
```

main.py

Run

Naan mudhalvan

```python
students =[
  Student("mani","A123",7.8),
  Student("ram","A124",9.8),
  Student("vik","A628",8.8),
  Student("raj","A383",8.2)
]

sorted_students=sort_students(students)

for student in sorted_students:
    print("Name: {},Roll Number: {},CGPA: {}".format(student.name,student.roll_number,student.cgpa))
```

main.py

Run

Naan mudhalvan

```
> python3 main.py
Name: ram,Roll Number:A124,CGPA: 9.8
Name: vik,Roll Number:A628,CGPA: 8.8
Name: raj,Roll Number:A383,CGPA: 8.2
Name: mani,Roll Number:A123,CGPA: 7.8
>
```

Console

Run

```python
def linearSearchProduct(productList, targetProduct):
  indices = []

  for index, product in enumerate(productList):
    if product == targetProduct:
      indices.append(index)

  return indices

products = ["shoes", "boot", "loafer", "shoes", "sandal", "shoes"]
target = "shoes"
target2 = 'apple'
result = linearSearchProduct(products, target)
result2 = linearSearchProduct(products, target)
print(result)
```

Ln 15, Col 14 History

main.py

Run

Python (9)

Exi

```
python3 main.py
[0, 3, 5]
```

Console

Run