```
#1)
def retVal(n):
        if len(n)>=3:
                for i in range(len(n)):
                        if i == 0 and (n[i].isnumeric()==False and
n[i]!='+'):
                                return False
                        elif i!= 0 and n[i].isnumeric()==False:
                                return False
                return True
        else:
                return False
#####################################################################
def regCall(clients,calls):
        origin = input(' Telefone origem? ')

        while retVal(origin)==False:
                origin = input(' Telefone origem? ')

        destination = input(' Telefone destino? ')
        while retVal(destination)==False:
                destination = input(' Telefone destino? ')

        duration = input(' Duraçao (s)? ')

        if origin not in clients:
                clients.append(origin)

        if origin not in calls:
                calls[origin] = (destination+'-'+duration+';')
        else:
                calls[origin]+=(destination+'-'+duration+';')

        return
#####################################################################
def fileRead(clients,calls):
        fileName = input('Ficheiro? ')
        try:
                fileOp = open(fileName,'r')

                for line in fileOp:
                        line = line.split('\t')

                        origin = line[0]
                        destination = line[1]
                        duration = line[2]

                        if origin not in clients:
                                clients.append(origin)

                        if origin not in calls:
                                calls[origin] = (destination+'-
'+duration.strip('\n')+';')
                        else:
                                calls[origin]+=(destination+'-
'+duration.strip('\n')+';')
```

```
                fileOp.close()

        except FileNotFoundError:
                fileRead(clients,calls)
#####################################################################
def lstClients(clients):
        strClients = ""
        clients = sorted(clients)
        for i in clients:
                strClients+=i+' '
        return strClients
#####################################################################
def bill(calls):
        client = input('Cliente? ')
        total = 0
        if client not in calls:
                print('Client not found')
                return()
        else:
                print('Fatura do cliente ',client)
                print('Destino\t\t\tDuraçao(s)\t\t\tCusto')
                clientCalls = calls[client].split(';')
                del clientCalls[-1]
                for call in clientCalls:
                        destination = call.rstrip('').split('-')[0]
                        duration = call.rstrip('').split('-')[1]

                        if destination[0] == '2':
                                price = 0.02*(int(duration)/60)
                        elif destination[0] == '+':
                                price = 0.80*(int(duration)/60)
                        elif destination[0]==client[0] and
destination[1]==client[1]:
                                price = 0.04*(int(duration)/60)
                        else:
                                price = 0.10*(int(duration)/60)
                        price = float("{0:.2f}".format(price))
                        total+=price

print('{}\t\t\t{}\t\t\t\t{}\n'.format(destination,duration,price))
                print('\t\t\t\t\tTotal:\t\t\t\t{}\n'.format(total))

print('Iniciado')
clients = []
calls = {}
while True:
        op = input("\n1) Registar chamada\n2) Ler ficheiro\n3) Listar
clientes\n4) Fatura\n5) Terminar\n\n0) Opçao ? ")
        if op=='1':
                regCall(clients,calls)

        elif op=='2':
                fileRead(clients,calls)

        elif op=='3':
                print(lstClients(clients))
```

```
        elif op=='4':
                bill(calls)

        elif op=='5':
                break
        else:
                print('Opçao Invalida\n')

print('\nTerminado')

#Notes: I chose to make it so all the calls get registered in he Calls
Dictionary/
#       the origin only shows up once and each calls
duration/destination gets added
#       as a string in the form of: Destination '-' duration ';'
```