

GAU - Gestão da Avaliação na UAL

Trabalho de Algoritmia e Programação

Projecto Fase 1

pAULO ENES DA SILVEIRA

Bruno Saraiva, 20160782; Rodrigo Araújo; 30000519 | Turma B; Diurno | 17/12/17

Indíce

1. Introdução-----------------------------------------------------------------------------------2
2. Estruturas de Dados Utilizados--------------------------------------------------------3
3. Elaboração e Justificação do Código--------------------------------------------4-11
4. Manual do Utilizador----------------------------------------------------------------12-16
5. Conclusão-----------------------------------------------------------------------------17-18
   1. Objetivos alcançados
   2. Dificuldades
   3. Como nos organizámos?
6. Listagem do Programa Fonte--------------------------------------------------19-187

Introdução

Este trabalho foi realizado no âmbito da disciplina de Algoritmia e Programação e tem como objetivo realizar um programa, em linguagem Python, que calcule e demonstre numa tabela e num gráfico as notas finais de uma ou várias disiciplinas. Para tal serão introduzidas as diversas notas intermédias (primeiro teste, segundo teste, primeiro trabalho, etc.), as quais mediante um peso percentual pré-definido, calcularão automaticamente a média ponderada que corresponderá à nota final da respectiva disciplina.

Para guardar as informações sobre as ponderações utilizamos as opereções de read fich e write fich para a gravação dos dados e abertura do ficheiro bem como a biblioteca pickle, para a transformação dos dados introduzidos no programa em linguagem Python para um ficheiro texto.

Estruturas de Dados Utilizados

Na ciência da computação, uma estrutura de dados é um modo particular de armazenamento e organização de dados em um computador de modo que possam ser usados eficientemente, facilitando assim a sua busca e modificação.

As estruturas de dados dividem-se em homogêneos (matrizes) e heterogêneos (registros):

- Estruturas homogêneas (matrizes):

Utilizadas para a construção das tabelas,ou seja, são conjuntos de dados formados pelo mesmo tipo de dado primitivo. Por exemplo:

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

- Estruturas heterogénias (registros):

As listas e a função date.time são registros pois são conjuntos de dados formados por tipos de dados diferentes. Por exemplo:

curso="Informática de Gestão"

disciplinas=["AP","NFC","IE","MAT1","IG"]

docentes=["Helena Curto","Rui Neves","Patricia Ferreira","Vera Pedragosa","Ana Quaresma","João Vela Bastos","Paulo Enes da Silveira","Marco Costa","Arlindo Donário","Ricardo Borges"]

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

Elaboração e Justificação do Código

1. Esta parte do programa consiste basicamente na importação das bibliotecas Python e a atribuição das listas (curso, disciplinas e docentes).

import heapq

import datetime

import numpy as np

import matplotlib.pyplot as plt

import pickle

curso="Informática de Gestão"

disciplinas=["AP","NFC","IE","MAT1","IG"]

docentes=["Helena Curto","Rui Neves","Patricia Ferreira","Vera Pedragosa","Ana Quaresma","João Vela Bastos","Paulo Enes da Silveira","Marco Costa","Arlindo Donário","Ricardo Borges"]

1. r=input("Escolha o que deseja observar [NFC/AP/IE/IG/MAT1/ALL]: ")

if r=="AP":

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

O que se pretende com este módulo é o seguinte:

Depois de se introduzir um valor por via teclado na questão *“Escolha o que deseja observar [NFC/AP/IE/IG/MAT1/ALL]:"* se a resposta for “AP” então a condição, *if r==”AP”, “*verifica-se” e os passos que se encontram “dentro” desta condição vão ser executados. Caso contrário, “*else:”,* o programa vai atribuir a *t1*, um “espaço vazio”, *t1=" "*.

Quando o programa compara se a string “AP” é a resposta, *r==”AP”*, então o primeiro passo que este vai executar é uma outra instrução que consiste em digitar a nota de um certo elemento de avaliação por via teclado, *t1=float(input("Nota do primeiro teste (AP): "))*, temos a instrução *float* porque as notas dos testes englobam números decimais, seguindo assim para o próximo passo da condição que é a comparação da varíavel *t1*, que foi introduzida por via teclado, se é maior que 20 ou menor que 0, ou seja basta o número introduzido ser maior que 20 ou menor que 0 e recebemos a mensagem que o valor que se introduziu está errado e força a instrução para inserir novamente o valor de *t1*, ou seja, temos de voltar a introduzir a nota do elemento de avaliação, só sai do ciclo while quando o valor t1 estiver contido entre 0 e 20.

Quando r é igual a NFC, AP, IE, IG ou MAT1 o programa executará sempre este módulo, mudando somente frases como por exemplo, “*Nota do primeiro trabalho (IE):”*.

Como se pode observar em baixo:

If r== “NFC”

(……….)

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

1. d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

Após a atribuição de todas as variáveis relativas aos elementos de avaliação, de cada respetiva disciplina, por exemplo *t1*, referido no ponto 2, vai ser possível a atribuição da data a que cada elemento de avalição

se vai realizar ou realizou, sendo assim possível a sua demonstração na tabela.

1. if t1==" " or t2==" " or t3==" " or p==" ":

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p," "]]

Caso não se saiba a nota de um dos elementos de avaliação então o programa executará o que se encontra dentro da condição, *if t1==" " or t2==" " or t3==" " or p==" ":* que só é possível se anteriormente for atribuido a *t1* ou *t2* ou a *p* um valor desconhecido.“ “.Posto isto o programa irá criar uma tabela a partir da matriz definida em AP, *for z in AP*. A operação *print()* é somente por conveniência ao utilizador, pois serve para criar um espaço entre a última instrução, que em maior parte dos casos seria a entrada por via teclado da nota da participação, e a tabela.

1. Para a tabela ficar organizada praticamente o que tivemos de fazer foi obeservar qual seria o maior elemento de cada coluna subtraindo o seu comprimento ao de, por exemplo, *" "\*10*.

Nas listas criadas, logo nas primeiras linhas de código foi-se buscar o curso e os respetivos docentes e assim estes dados imprimidos na tabela. Como temos dados insuficientes, na tabela a média final não foi calculada recebendo assim um aviso na consola de que os dados não são suficientes para ilustrar o gráfico. *print("\nNOTA: Dados insuficientes para ilustrar o gráfico")*

print()

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|")

print("\nCorpo docente do curso de",curso,"referente à disciplina de {0}: {1} e {2}".format(disciplinas[0],docentes[6],docentes[7]))

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

1. Caso contrário, *else*, sabendo todos os valores dos elementos de avaliação de uma dada disciplina já é possível calcular a média final e assim ilustrar o gráfico e os valores na tabela.

else:

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

Através desta função cálcula-se a média final.

Pelo mesmo método anteriormento explicado cria-se a tabela, não recebendo a mensagem de aviso pela falta de dados.

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

print()

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|")

print("\nCorpo docente do curso de",curso,"referente à disciplina de {0}: {1} e {2}".format(disciplinas[0],docentes[6],docentes[7]))

1. Desta vez não temos informação desconhecida sendo assim possível a ilustração do gráfico de barras na vertical.

x=["Teste1","Trabalho","Teste2","Participação","Final"]

y=[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em AP")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

1. Se a escolha fosse ALL então o programa irá pedir todas as notas de cada disciplina e calculando passo a passo a média de cada disciplina e no fim compilar tudo num gráfico de barras horizontais, que serão os resultados finais de todas as disciplinas do primeiro semestre.

x=[round(i(n1,n2,q),2),round(k(e1,e2,e3,h),2),round(a(MT,TI,TF,v),2),round(f(t1,t2,t3,p),2),round(j(g1,g2,u),2)]

Utilizou-se a operação round pois as notas finais são sempre arredondadas.

y=["NFC","IE","MAT1","AP","IG"]

def plotgraph (y,x):

plt.xlabel("Notas finais")

plt.ylabel("Disciplinas")

plt.title("Classificações finais de todas as disciplinas")

plt.barh(y,x)

plt.xlim(0,20)

plt.xticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(y,x)

else:

print()

A operação plt.barh(y,x) é o que permite a ilustrção de um gráfico de barras horizontal e a operação plt.bar(y,x) permite a criação de um gráfico de barras vertical.

1. Operação que escreve os seguintes caractéres como cabeçalho do ficheiro texto. Fich1=open(“GAU”,”w”) é a operação que vai atribuir o título ao ficheiro e permitir a sua edição.

fich1=open("GAU","w")

fich1.write("Sistema de Avaliação GAU")

print("\nAvaliação contínua 1ºano/1ºsemestre | Curso: {0}".format(curso),file=fich1)

1. Dos vários dados introduzidos no programa somente os mais relevantes, ou seja, as ponderações, disciplinas, docentes, datas e notas finais foi necessária a utilização da operação file=fich1.

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

1. Para fechar o modo de edição do ficheiro utilizámos o fich1.close(), de forma a gravação do ficheiro ser possível através de uma função com a operação read (r).

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

1. Para serializar os dados em binário foi necessário importar a biblioteca pickle (import pickle) para a utilização das funções de conversão. Evoca-se o nome do ficheiro e de seguida a operação lerAfixaFich(nomeFich) para criar um novo ficheiro, utilizando as listas do original. Abre-se o ficheiro com o comando with open('fich.pkl','wb') as f: para começar a editar os dados em binário, write binary (wb). Como explicado anteriormente, parágrafo 4, se a resposta for um valor vazio, desconhece-se uma nota de um dos elemento de avaliação, temos que converter as listas AP, docentes e disciplinas em binário, pois são estas que aparecem como output quando não se sabe um dos valores dos elementos de avaliação.

Caso se conheçam todos os valores de todos os elementos de avaliação de uma determinada disciplina, neste caso AP, temos que serializar todas as listas que aparecem no output bem como o gráfico, AP, docentes, disciplinas, eixo x, eixo y.

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

fich.close()

1. Abrimos o ficheiro que contém as variáveis e listas serializadas em binário com a operação with open('fich.pkl','rb') as f: que vai permitir a leitura, ou seja vai descodificar a linguagem transcrita em binário para o seu valor original e pela mesma lógica explicada no parágrafo anterior (12) temos de utilizar as variáves e listas que queremos descodificar, ou seja, utilizamos APread=pickle.load(f) para dar inicío à descodificação.

Também temos que atribuir as variaves a operação docentesread=pickle.load(f) para descodificar o valor indicado, neste caso a lista docentes.

with open('fich.pkl','rb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

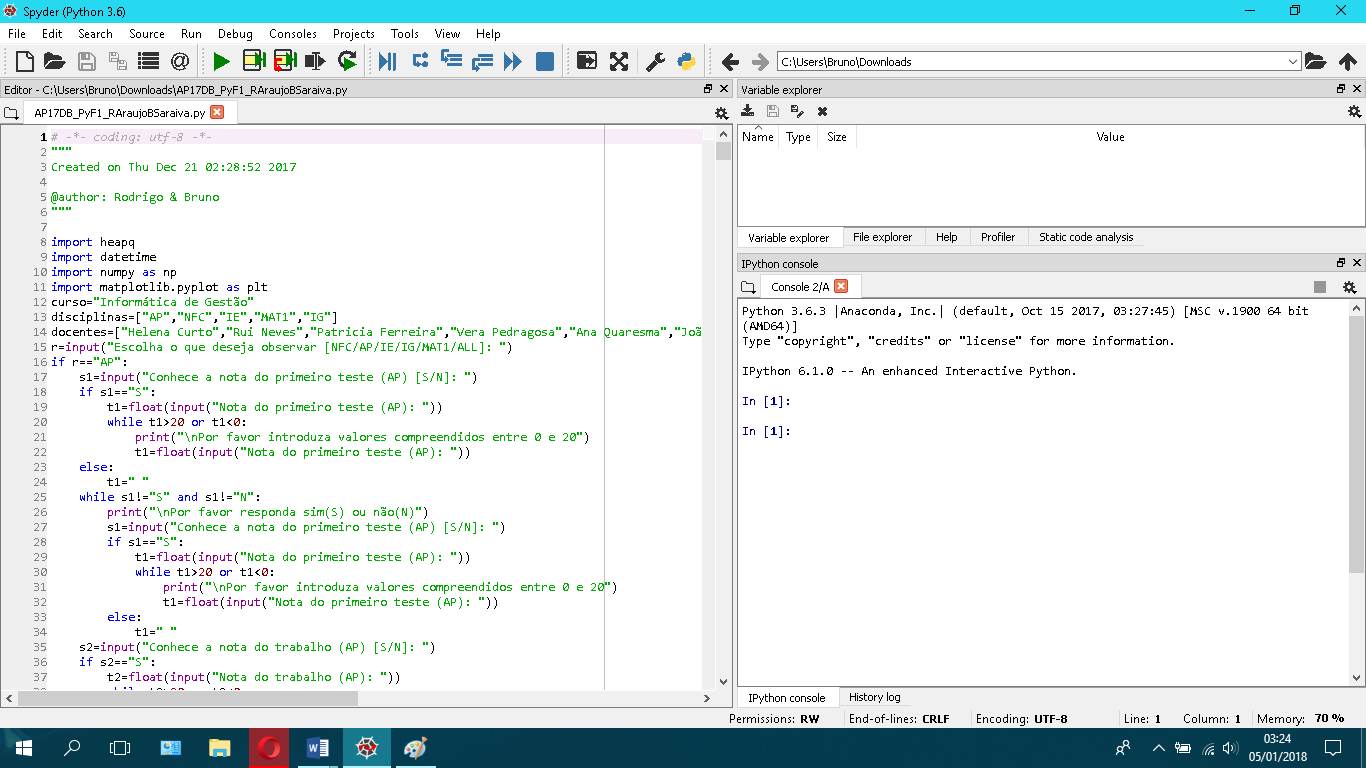
print(xread)

print()

print(yread)

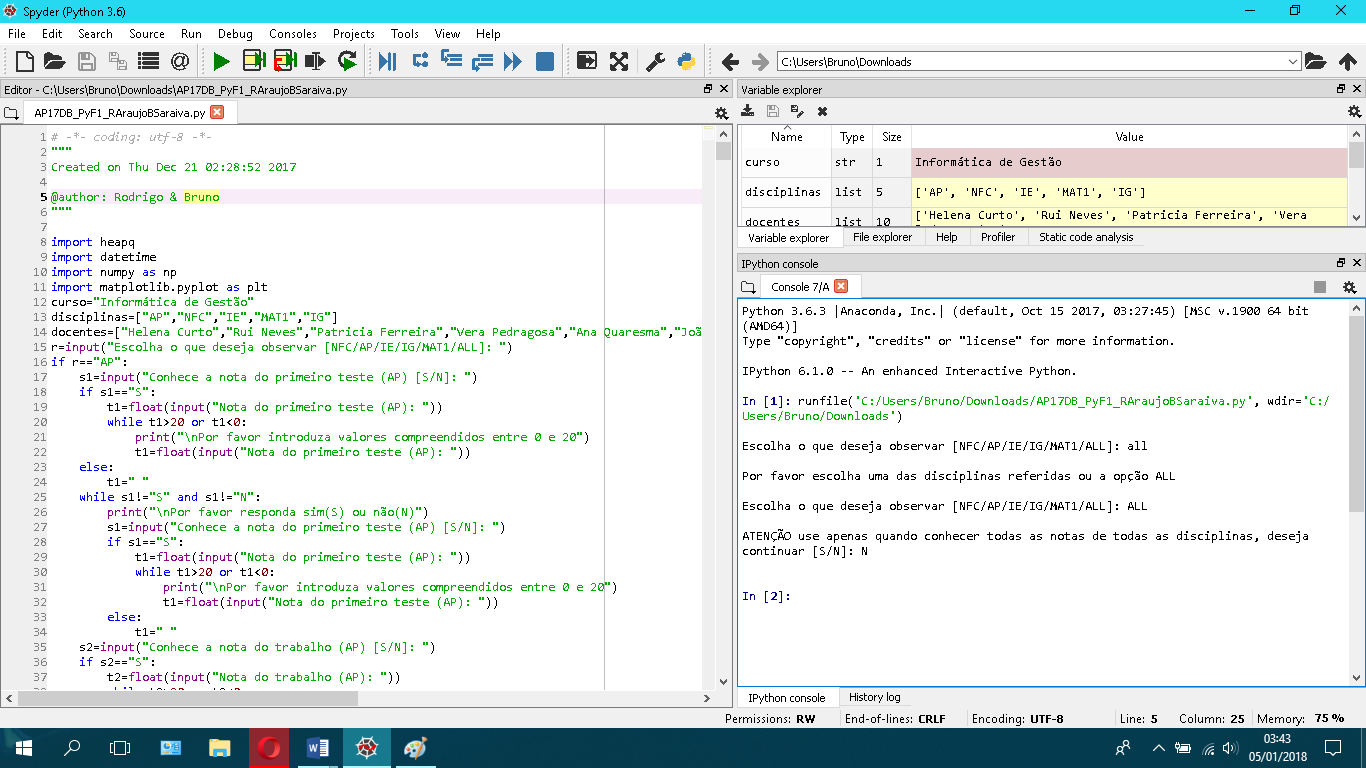
f.close()

Manual do Utilizador



**1**

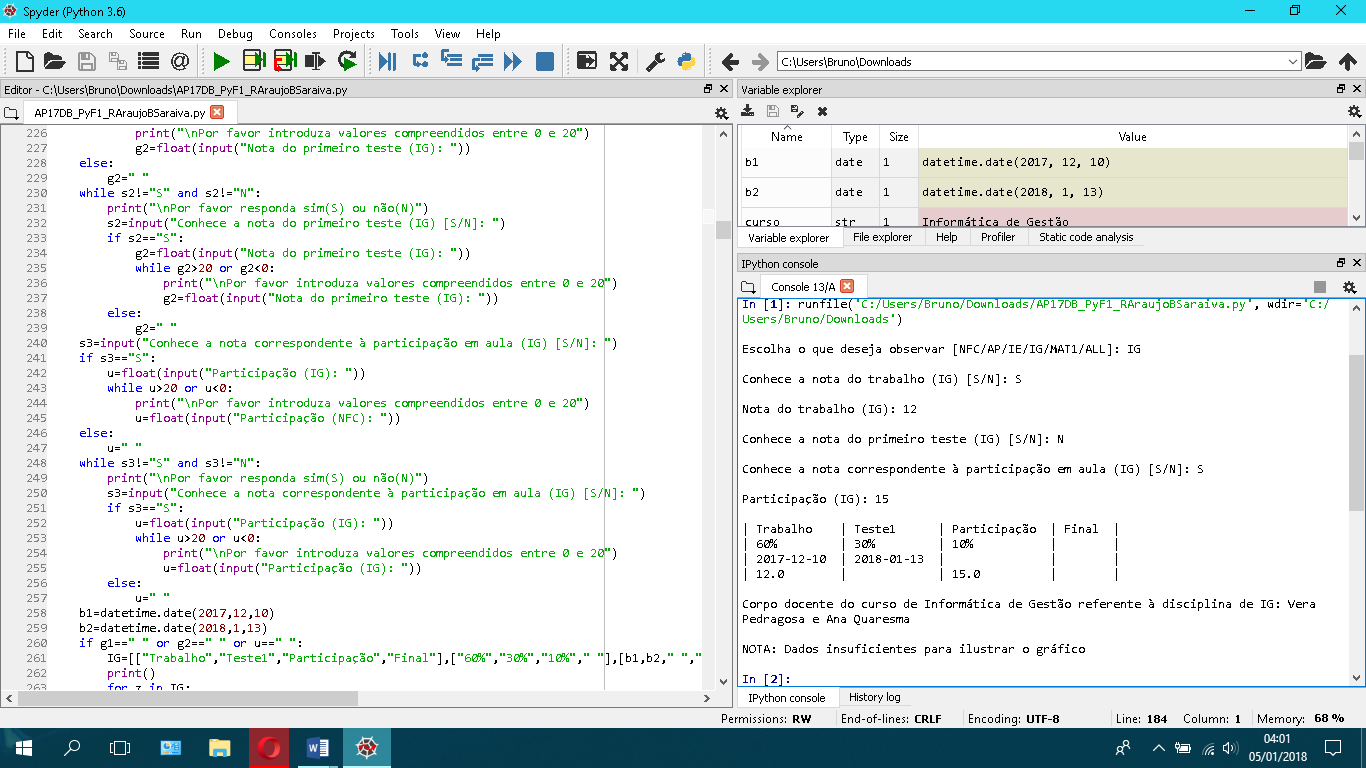
1. Inicializar o programa clicando na seta verde, como demonstrado na figura, ou se preferir carregue na tecla F5;



**2**

1. Introduza na consola, que se encontra no canto inferior direito na janela do Spyder, o que pretende observar.

O programa só aceita letras em maiuscula e valores corretos, o programa pede somente [NFC/AP/IE/IG/MAT1/ALL] e desta forma terá de se introduzir como pedido. Ter em atenção que escolher a opção ALL, ou seja todas as disciplinas, só funciona se todas as notas de todos os elementos de avaliação forem conhecidas, caso contrário o programa acaba. Por essa raão recebe a mensagem de aviso a perguntar se quer continuar, se responder com N o programa acaba e terá de voltar a inicializar o programa como está explicado no passo (1);

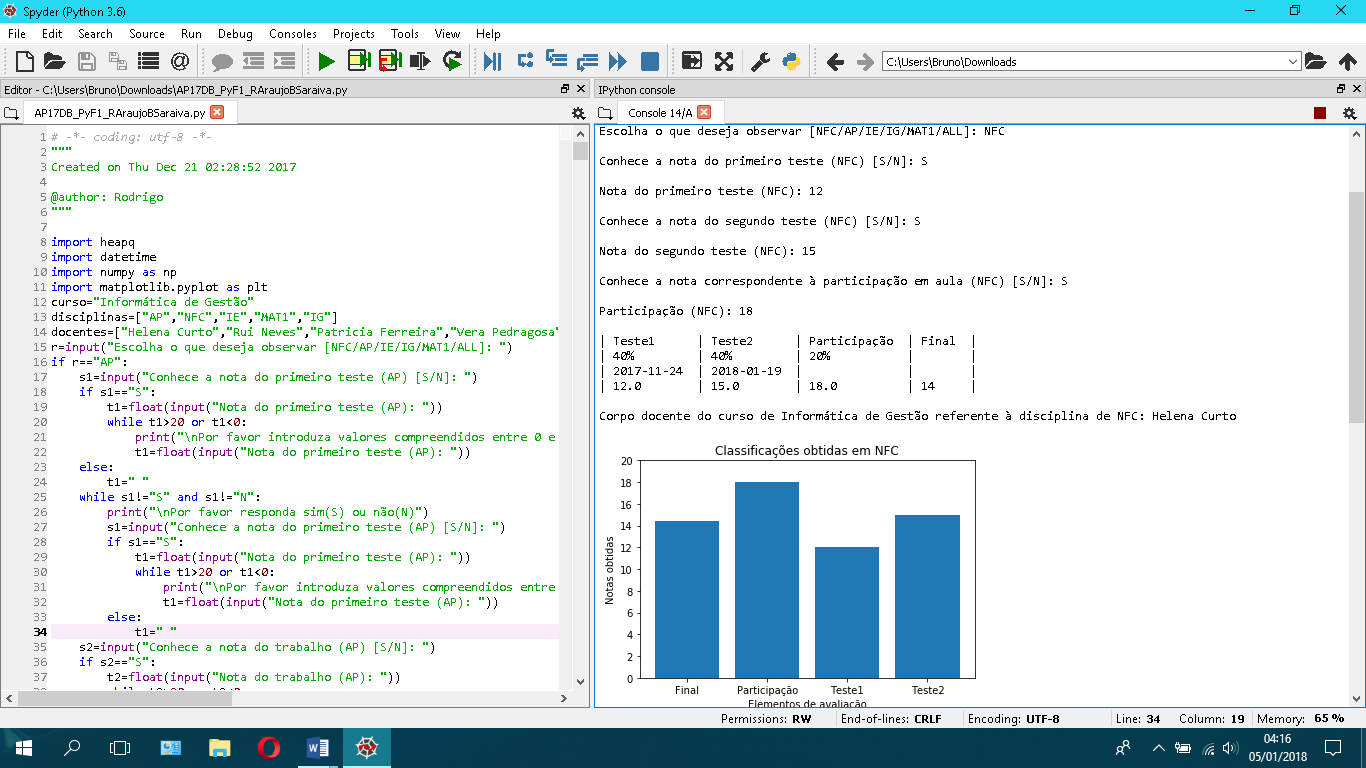


**3**

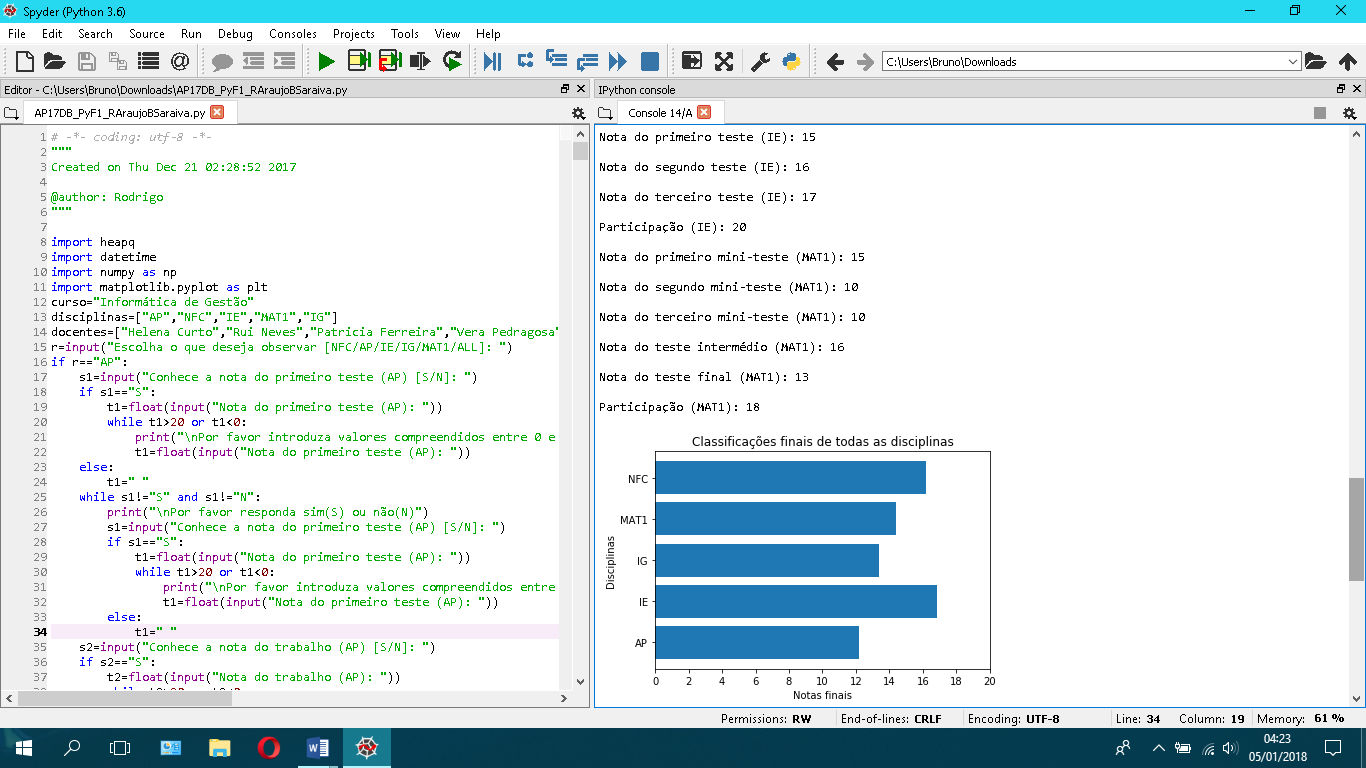
1. Outro fator a ter em atenção quando introduz os dados é que se não conhecer uma ou mais notas esta não aparecerá na tabela, assim como também não aparecerá a nota final pois o cálculo da média é impossível. Também receberá uma nota de aviso a informar que os dados foram insuficientes para a ilustração do gráfico.

**Se seguiu corretamente as instruções poderá obter resultados parecidos a alguns destes dois exemplos:**

**Exemplo 1, se escolheu apenas uma disciplina:**

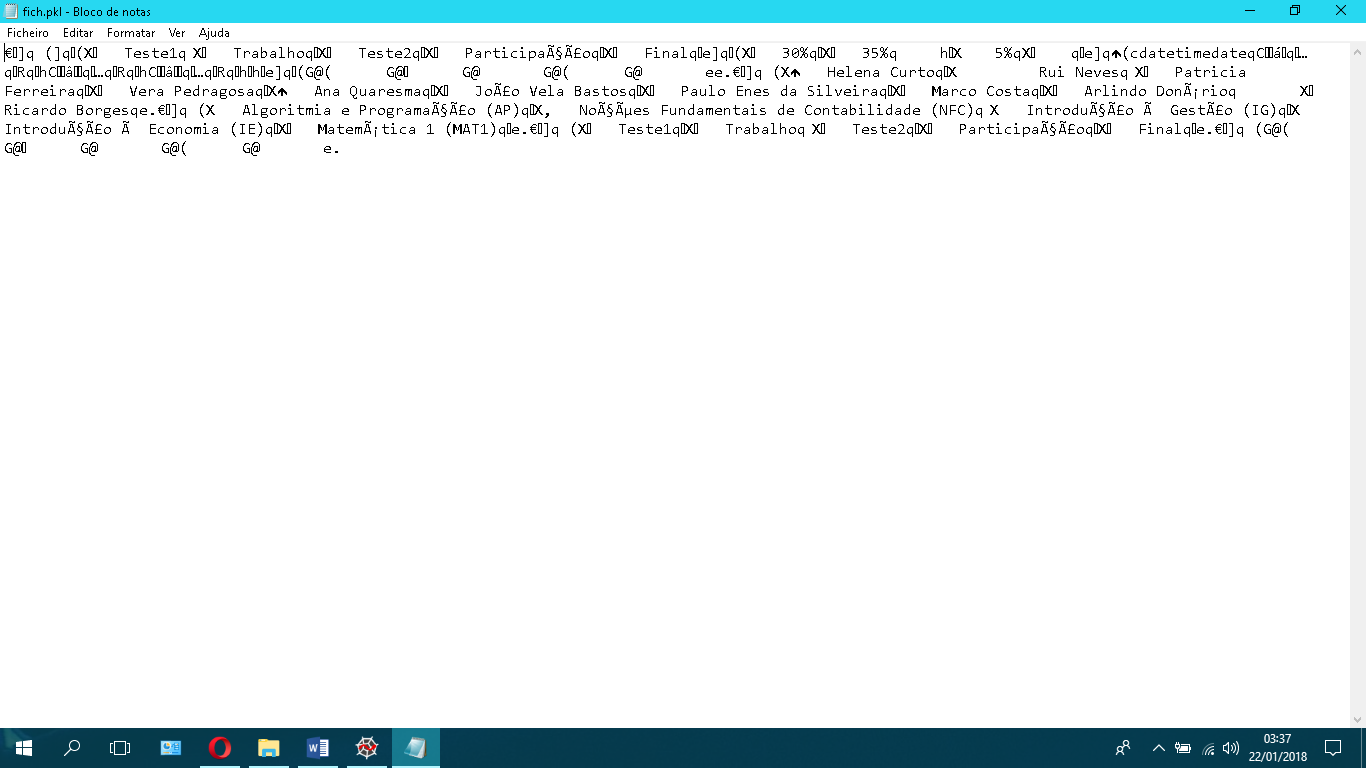


**Exemplo 2, se escolheu a opção ALL:**



1. Finalmente ter atenção onde guardou o ficherio Python pois é nessa pasta que se vai encontrar o ficheiro texto com os dados de maior importância relativamente às ponderações, bem como se encontrará também um ficheiro texto com o mesmo contéudo mas serializado em binário. Para observer o contéudo dos ficheiros texto basta abrir com o bloco de notas.

Eis o contéudo do ficheiro texto serializado:



Conclusão

Com a realização deste trabalho adquirimos mais conhecimento relativamente à programação em linguagem Python e quão extenso se pode tornar um simples programa para calcular médias finais e as exibir numa tabela, bem como salvar os dados relativamente importantes num ficheiro e a opção de se conseguir serializar e des-serializar o ficheiro em texto.

Muitas das dificuldades que nós tivemos que nos deparar foram as seguintes:

- Construção das tabelas, mas depois de um pouco de pesquisa fomos capazes de ultrapassar esta dificuldade facilmente tomando como base este vídeo [https://www.youtube.com/watch?v=B9BRuhqEb2Q](https://www.youtube.com/watch?v=B9BRuhqEb2Q%20) e adaptando o seu conteúdo ao que nos era pedido para o trabalho.

- Resultado final dos gráficos, pois estes não adquiriam os valores que nós pretendiamos ou quando adquiriam eram apresentados de um modo que não correspondia às nossas expectativas, mas ao rever a “Aula 6 Gráficos e Matrizes” e ao complementar a informação que retiramos da Aula 6 com alguns passos provenientes da internet conseguimos superar esta dificuldade.

Todos os objectivos foram alcançados à excepção de um objetivo que é que encontra explícito em baixo…

Infelizmente não conseguimos calcular e ler, no nosso programa, quanto é que o aluno teria de obter no último momento de avaliação para poder ter a nota miníma de 10 valores, ou seja, para poder ser aprovado à disciplina. Várias foram as tentativas mas mesmo assim não nos foi possível encontrar uma solução.

Como nos organizámos e desenvolvemos o trabalho?

Recrorrendo ao guião fornecido pelo professor, decidimos priorizar somente a construção do ficheiro em Python, deixando assim o relatório para o fim.

Utilizámos um programa de vídeo-chamada que permite a partilha de ecrã e de documentos tornando assim eficaz o trabalho em grupo, pois tornou fácil a partilha de ideias pelo simples facto de o contacto ser sempre possível quer estando em casa ou não.

Começamos a definir os objetivos que tinham de ser alcançados e o tempo que demorariam a ser atingidos.

Após verificar que o ficheiro em linguagem Python estava funcional, começamos a realização do relatório.

Relativamente à divisão do trabalho pelos membros do grupo, começamos ambos a trabalhar no código em Python e quando chegamos a cerca de 25% do programa pretendido acordamos que os restantes 75% do programa, seriam finalizados por um membro do grupo. No relatório a situação foi igual mas desta vez foi o colega, que eleborou até aos 25% do programa em Python, que fez os 75% do relátorio ficando os restantes 25% para o colega que fez 75% do programa em linguagem em Python.

Listagem do Programa Fonte

import heapq

import datetime

import numpy as np

import matplotlib.pyplot as plt

import pickle

curso="Informática de Gestão"

disciplinas=["Algoritmia e Programação (AP)","Noções Fundamentais de Contabilidade (NFC)","Introdução à Gestão (IG)","Introdução à Economia (IE)","Matemática 1 (MAT1)"]

docentes=["Helena Curto","Rui Neves","Patricia Ferreira","Vera Pedragosa","Ana Quaresma","João Vela Bastos","Paulo Enes da Silveira","Marco Costa","Arlindo Donário","Ricardo Borges"]

fich1=open("GAU","w")

fich1.write("Sistema de Avaliação GAU")

print("\nAvaliação contínua 1ºano/1ºsemestre | Curso: {0}".format(curso),file=fich1)

r=input("Escolha o que deseja observar [NFC/AP/IE/IG/MAT1/ALL]: ")

if r=="AP":

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

if t1==" " or t2==" " or t3==" " or p==" ":

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p," "]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Trabalho","Teste2","Participação","Final"]

y=[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em AP")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="NFC":

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

if n1==" " or n2==" " or q==" ":

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q," "]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Teste1","Teste2","Participação","Final"]

y=[n1,n2,q,round(i(n1,n2,q),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em NFC")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if n1==" " or n2==" " or q==" ":

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if n1==" " or n2==" " or q==" ":

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IG":

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

else:

u=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (IG): "))

else:

u=" "

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

if g1==" " or g2==" " or u==" ":

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u," "]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Trabalho","Teste1","Participação","Final"]

y=[g1,g2,u,round(j(g1,g2,u),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IG")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if g1==" " or g2==" " or u==" ":

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if g1==" " or g2==" " or u==" ":

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IE":

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

if e1==" " or e2==" " or e3==" " or h==" ":

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h," "]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Teste2","Teste3","Participação","Final"]

y=[e1,e2,e3,h,round(k(e1,e2,e3,h),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IE")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="MAT1":

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

while s5!="S" and s5!="N":

print("\nPor favor responda sim(S) ou não(N)")

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

while s6!="S" and s6!="N":

print("\nPor favor responda sim(S) ou não(N)")

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[" ",TI,TF,v," "]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["m1","m2","m3","MT","TI","TF","Partipação","Final"]

y=[m1,m2,m3,MT,TI,TF,v,round(a(MT,TI,TF,v),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em MAT1")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(tm,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

else:

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

tmread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(tmread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="ALL":

all1=input("ATENÇÃO use apenas quando conhecer todas as suas classificações, deseja continuar [S/N]: ")

if all1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

print()

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

print()

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

print()

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

print()

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=[round(i(n1,n2,q),2),round(k(e1,e2,e3,h),2),round(a(MT,TI,TF,v),2),round(f(t1,t2,t3,p),2),round(j(g1,g2,u),2)]

y=["NFC","IE","MAT1","AP","IG"]

def plotgraph (y,x):

plt.xlabel("Notas finais")

plt.ylabel("Disciplinas")

plt.title("Classificações finais de todas as disciplinas")

plt.barh(y,x)

plt.xlim(0,20)

plt.xticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(y,x)

print()

else:

r=input("Então, escolha o que deseja observar [NFC/AP/IE/IG/MAT1]: ")

if r=="AP":

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

if t1==" " or t2==" " or t3==" " or p==" ":

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p," "]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Trabalho","Teste2","Participação","Final"]

y=[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em AP")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="NFC":

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

if n1==" " or n2==" " or q==" ":

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q," "]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Teste1","Teste2","Participação","Final"]

y=[n1,n2,q,round(i(n1,n2,q),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em NFC")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if n1==" " or n2==" " or q==" ":

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if n1==" " or n2==" " or q==" ":

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IG":

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

else:

u=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (IG): "))

else:

u=" "

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

if g1==" " or g2==" " or u==" ":

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u," "]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Trabalho","Teste1","Participação","Final"]

y=[g1,g2,u,round(j(g1,g2,u),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IG")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if g1==" " or g2==" " or u==" ":

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if g1==" " or g2==" " or u==" ":

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IE":

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

if e1==" " or e2==" " or e3==" " or h==" ":

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h," "]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Teste2","Teste3","Participação","Final"]

y=[e1,e2,e3,h,round(k(e1,e2,e3,h),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IE")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="MAT1":

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

while s5!="S" and s5!="N":

print("\nPor favor responda sim(S) ou não(N)")

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

while s6!="S" and s6!="N":

print("\nPor favor responda sim(S) ou não(N)")

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[" ",TI,TF,v," "]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["m1","m2","m3","MT","TI","TF","Partipação","Final"]

y=[m1,m2,m3,MT,TI,TF,v,round(a(MT,TI,TF,v),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em MAT1")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(tm,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

else:

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

tmread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(tmread)

print()

print(xread)

print()

print(yread)

f.close()

while all1!="S" and all1!="N":

all1=input("ATENÇÃO use apenas quando conhecer todas as suas classificações, deseja continuar [S/N]: ")

if all1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

print()

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

print()

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

print()

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

print()

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=[round(i(n1,n2,q),2),round(k(e1,e2,e3,h),2),round(a(MT,TI,TF,v),2),round(f(t1,t2,t3,p),2),round(j(g1,g2,u),2)]

y=["NFC","IE","MAT1","AP","IG"]

def plotgraph (y,x):

plt.xlabel("Notas finais")

plt.ylabel("Disciplinas")

plt.title("Classificações finais de todas as disciplinas")

plt.barh(y,x)

plt.xlim(0,20)

plt.xticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(y,x)

print()

else:

r=input("Então, escolha o que deseja observar [NFC/AP/IE/IG/MAT1]: ")

if r=="AP":

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

if t1==" " or t2==" " or t3==" " or p==" ":

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p," "]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Trabalho","Teste2","Participação","Final"]

y=[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em AP")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="NFC":

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

if n1==" " or n2==" " or q==" ":

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q," "]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Teste1","Teste2","Participação","Final"]

y=[n1,n2,q,round(i(n1,n2,q),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em NFC")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if n1==" " or n2==" " or q==" ":

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if n1==" " or n2==" " or q==" ":

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IG":

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

else:

u=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (IG): "))

else:

u=" "

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

if g1==" " or g2==" " or u==" ":

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u," "]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Trabalho","Teste1","Participação","Final"]

y=[g1,g2,u,round(j(g1,g2,u),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IG")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if g1==" " or g2==" " or u==" ":

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if g1==" " or g2==" " or u==" ":

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IE":

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

if e1==" " or e2==" " or e3==" " or h==" ":

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h," "]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Teste2","Teste3","Participação","Final"]

y=[e1,e2,e3,h,round(k(e1,e2,e3,h),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IE")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="MAT1":

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

while s5!="S" and s5!="N":

print("\nPor favor responda sim(S) ou não(N)")

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

while s6!="S" and s6!="N":

print("\nPor favor responda sim(S) ou não(N)")

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[" ",TI,TF,v," "]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["m1","m2","m3","MT","TI","TF","Partipação","Final"]

y=[m1,m2,m3,MT,TI,TF,v,round(a(MT,TI,TF,v),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em MAT1")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(tm,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

else:

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

tmread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(tmread)

print()

print(xread)

print()

print(yread)

f.close()

fich1.close()

if all1=="S":

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

pickle.dump(AP,f)

pickle.dump(NFC,f)

pickle.dump(IG,f)

pickle.dump(IE,f)

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

with open('fich.pkl','rb') as f:

APread=pickle.load(f)

NFCread=pickle.load(f)

IGread=pickle.load(f)

IEread=pickle.load(f)

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(APread)

print()

print(NFCread)

print()

print(IGread)

print()

print(IEread)

print()

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

while r!="NFC" and r!="MAT1" and r!="AP" and r!="IE" and r!="IG" and r!="ALL":

print("\nPor favor escolha uma das disciplinas referidas ou a opção ALL")

r=input("Escolha o que deseja observar [NFC/AP/IE/IG/MAT1/ALL]: ")

if r=="AP":

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

if t1==" " or t2==" " or t3==" " or p==" ":

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p," "]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Trabalho","Teste2","Participação","Final"]

y=[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em AP")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="NFC":

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

if n1==" " or n2==" " or q==" ":

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q," "]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Teste1","Teste2","Participação","Final"]

y=[n1,n2,q,round(i(n1,n2,q),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em NFC")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if n1==" " or n2==" " or q==" ":

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if n1==" " or n2==" " or q==" ":

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IG":

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

else:

u=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (IG): "))

else:

u=" "

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

if g1==" " or g2==" " or u==" ":

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u," "]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Trabalho","Teste1","Participação","Final"]

y=[g1,g2,u,round(j(g1,g2,u),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IG")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if g1==" " or g2==" " or u==" ":

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if g1==" " or g2==" " or u==" ":

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IE":

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

if e1==" " or e2==" " or e3==" " or h==" ":

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h," "]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Teste2","Teste3","Participação","Final"]

y=[e1,e2,e3,h,round(k(e1,e2,e3,h),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IE")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="MAT1":

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

while s5!="S" and s5!="N":

print("\nPor favor responda sim(S) ou não(N)")

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

while s6!="S" and s6!="N":

print("\nPor favor responda sim(S) ou não(N)")

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[" ",TI,TF,v," "]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["m1","m2","m3","MT","TI","TF","Partipação","Final"]

y=[m1,m2,m3,MT,TI,TF,v,round(a(MT,TI,TF,v),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em MAT1")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(tm,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

else:

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

tmread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(tmread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="ALL":

all1=input("ATENÇÃO use apenas quando conhecer todas as suas classificações, deseja continuar [S/N]: ")

if all1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

print()

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

print()

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

print()

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

print()

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=[round(i(n1,n2,q),2),round(k(e1,e2,e3,h),2),round(a(MT,TI,TF,v),2),round(f(t1,t2,t3,p),2),round(j(g1,g2,u),2)]

y=["NFC","IE","MAT1","AP","IG"]

def plotgraph (y,x):

plt.xlabel("Notas finais")

plt.ylabel("Disciplinas")

plt.title("Classificações finais de todas as disciplinas")

plt.barh(y,x)

plt.xlim(0,20)

plt.xticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(y,x)

print()

else:

r=input("Então, escolha o que deseja observar [NFC/AP/IE/IG/MAT1]: ")

if r=="AP":

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

if t1==" " or t2==" " or t3==" " or p==" ":

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p," "]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Trabalho","Teste2","Participação","Final"]

y=[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em AP")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="NFC":

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

if n1==" " or n2==" " or q==" ":

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q," "]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Teste1","Teste2","Participação","Final"]

y=[n1,n2,q,round(i(n1,n2,q),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em NFC")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if n1==" " or n2==" " or q==" ":

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if n1==" " or n2==" " or q==" ":

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IG":

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

else:

u=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (IG): "))

else:

u=" "

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

if g1==" " or g2==" " or u==" ":

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u," "]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Trabalho","Teste1","Participação","Final"]

y=[g1,g2,u,round(j(g1,g2,u),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IG")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if g1==" " or g2==" " or u==" ":

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if g1==" " or g2==" " or u==" ":

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IE":

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

if e1==" " or e2==" " or e3==" " or h==" ":

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h," "]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Teste2","Teste3","Participação","Final"]

y=[e1,e2,e3,h,round(k(e1,e2,e3,h),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IE")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="MAT1":

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

while s5!="S" and s5!="N":

print("\nPor favor responda sim(S) ou não(N)")

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

while s6!="S" and s6!="N":

print("\nPor favor responda sim(S) ou não(N)")

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[" ",TI,TF,v," "]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["m1","m2","m3","MT","TI","TF","Partipação","Final"]

y=[m1,m2,m3,MT,TI,TF,v,round(a(MT,TI,TF,v),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em MAT1")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(tm,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

else:

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

tmread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(tmread)

print()

print(xread)

print()

print(yread)

f.close()

while all1!="S" and all1!="N":

all1=input("ATENÇÃO use apenas quando conhecer todas as suas classificações, deseja continuar [S/N]: ")

if all1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

print()

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

print()

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

print()

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

print()

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=[round(i(n1,n2,q),2),round(k(e1,e2,e3,h),2),round(a(MT,TI,TF,v),2),round(f(t1,t2,t3,p),2),round(j(g1,g2,u),2)]

y=["NFC","IE","MAT1","AP","IG"]

def plotgraph (y,x):

plt.xlabel("Notas finais")

plt.ylabel("Disciplinas")

plt.title("Classificações finais de todas as disciplinas")

plt.barh(y,x)

plt.xlim(0,20)

plt.xticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(y,x)

print()

else:

r=input("Então, escolha o que deseja observar [NFC/AP/IE/IG/MAT1]: ")

if r=="AP":

print("\nDisciplina: {0}".format(disciplinas[0]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[6],docentes[7]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (AP) [S/N]: ")

if s1=="S":

t1=float(input("Nota do primeiro teste (AP): "))

while t1>20 or t1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t1=float(input("Nota do primeiro teste (AP): "))

else:

t1=" "

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do trabalho (AP) [S/N]: ")

if s2=="S":

t2=float(input("Nota do trabalho (AP): "))

while t2>20 or t2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t2=float(input("Nota do trabalho (AP): "))

else:

t2=" "

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do segundo teste (AP) [S/N]: ")

if s3=="S":

t3=float(input("Nota do segundo teste (AP): "))

while t3>20 or t3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

t3=float(input("Nota do segundo teste (AP): "))

else:

t3=" "

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (AP) [S/N]: ")

if s4=="S":

p=float(input("Participação (AP): "))

while p>20 or p<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

p=float(input("Participação (AP): "))

else:

p=" "

d1=datetime.date(2017,11,28)

d2=datetime.date(2018,1,5)

d3=datetime.date(2018,1,16)

if t1==" " or t2==" " or t3==" " or p==" ":

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p," "]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def f (t1,t2,t3,p):

f=(t1\*0.30)+(t2\*0.35)+(t3\*0.30)+(p\*0.05)

return f

AP=[["Teste1","Trabalho","Teste2","Participação","Final"],["30%","35%","30%","5%"," "],[d1,d2,d3," "," "],[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]]

for z in AP:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Trabalho","Teste2","Participação","Final"]

y=[t1,t2,t3,p,round(f(t1,t2,t3,p),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em AP")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(AP,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if t1==" " or t2==" " or t3==" " or p==" ":

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

APread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(APread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="NFC":

print("\nDisciplina: {0}".format(disciplinas[1]),file=fich1)

print("Docente: {0}".format(docentes[0]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (NFC) [S/N]: ")

if s1=="S":

n1=float(input("Nota do primeiro teste (NFC): "))

while n1>20 or n1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n1=float(input("Nota do primeiro teste (NFC): "))

else:

n1=" "

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (NFC) [S/N]: ")

if s2=="S":

n2=float(input("Nota do segundo teste (NFC): "))

while n2>20 or n2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

n2=float(input("Nota do segundo teste (NFC): "))

else:

n2=" "

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (NFC) [S/N]: ")

if s3=="S":

q=float(input("Participação (NFC): "))

while q>20 or q<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

q=float(input("Participação (NFC): "))

else:

q=" "

c1=datetime.date(2017,11,24)

c2=datetime.date(2018,1,19)

if n1==" " or n2==" " or q==" ":

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q," "]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def i (n1,n2,q):

i=(n1\*0.40)+(n2\*0.40)+(q\*0.20)

return i

NFC=[["Teste1","Teste2","Participação","Final"],["40%","40%","20%"," "],[c1,c2," "," "],[n1,n2,q,round(i(n1,n2,q))]]

for z in NFC:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Teste1","Teste2","Participação","Final"]

y=[n1,n2,q,round(i(n1,n2,q),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em NFC")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if n1==" " or n2==" " or q==" ":

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(NFC,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if n1==" " or n2==" " or q==" ":

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

NFCread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(NFCread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IG":

print("\nDisciplina: {0}".format(disciplinas[2]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[3],docentes[4]),file=fich1)

print("Elementos de avaliação: 3",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do trabalho (IG) [S/N]: ")

if s1=="S":

g1=float(input("Nota do trabalho (IG): "))

while g1>20 or g1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g1=float(input("Nota do trabalho (IG): "))

else:

g1=" "

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do primeiro teste (IG) [S/N]: ")

if s2=="S":

g2=float(input("Nota do primeiro teste (IG): "))

while g2>20 or g2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

g2=float(input("Nota do primeiro teste (IG): "))

else:

g2=" "

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (NFC): "))

else:

u=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota correspondente à participação em aula (IG) [S/N]: ")

if s3=="S":

u=float(input("Participação (IG): "))

while u>20 or u<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

u=float(input("Participação (IG): "))

else:

u=" "

b1=datetime.date(2017,12,10)

b2=datetime.date(2018,1,13)

if g1==" " or g2==" " or u==" ":

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u," "]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def j (g1,g2,u):

j=(g1\*0.30)+(g2\*0.60)+(u\*0.10)

return j

IG=[["Trabalho","Teste1","Participação","Final"],["60%","30%","10%"," "],[b1,b2," "," "],[g1,g2,u,round(j(g1,g2,u))]]

for z in IG:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(12-len(str(z[2]))),"|",z[3]," "\*(5-len(str(z[3]))),"|",file=fich1)

x=["Trabalho","Teste1","Participação","Final"]

y=[g1,g2,u,round(j(g1,g2,u),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IG")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if g1==" " or g2==" " or u==" ":

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IG,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if g1==" " or g2==" " or u==" ":

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IGread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IGread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="IE":

print("\nDisciplina: {0}".format(disciplinas[3]),file=fich1)

print("Docentes: {0} e {1}".format(docentes[8],docentes[9]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro teste (IE) [S/N]: ")

if s1=="S":

e1=float(input("Nota do primeiro teste (IE): "))

while e1>20 or e1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e1=float(input("Nota do primeiro teste (IE): "))

else:

e1=" "

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo teste (IE) [S/N]: ")

if s2=="S":

e2=float(input("Nota do segundo teste (IE): "))

while e2>20 or e2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e2=float(input("Nota do segundo teste (IE): "))

else:

e2=" "

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro teste (IE) [S/N]: ")

if s3=="S":

e3=float(input("Nota do terceiro teste (IE): "))

while e3>20 or e3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

e3=float(input("Nota do terceiro teste (IE): "))

else:

e3=" "

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota correspondente à participação em aula (IE) [S/N]: ")

if s4=="S":

h=float(input("Participação (IE): "))

while h>20 or h<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

h=float(input("Participação (IE): "))

else:

h=" "

w1=datetime.date(2017,11,9)

w2=datetime.date(2017,12,7)

w3=datetime.date(2018,1,25)

if e1==" " or e2==" " or e3==" " or h==" ":

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h," "]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

def k (e1,e2,e3,h):

k=(e1\*0.18)+(e2\*0.30)+(e3\*0.35)+(h\*0.17)

return k

IE=[["Teste1","Teste2","Teste3","Participação","Final"],["18%","30%","35%","17%"," "],[w1,w2,w3," "," "],[e1,e2,e3,h,round(k(e1,e2,e3,h))]]

for z in IE:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["Teste1","Teste2","Teste3","Participação","Final"]

y=[e1,e2,e3,h,round(k(e1,e2,e3,h),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em IE")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(IE,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if e1==" " or e2==" " or e3==" " or h==" ":

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

else:

IEread=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(IEread)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)

f.close()

if r=="MAT1":

print("\nDisciplina: {0}".format(disciplinas[4]),file=fich1)

print("Docentes: {0}, {1} e {2}".format(docentes[1],docentes[2],docentes[5]),file=fich1)

print("Elementos de avaliação: 4",file=fich1)

print(file=fich1)

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

while s1!="S" and s1!="N":

print("\nPor favor responda sim(S) ou não(N)")

s1=input("Conhece a nota do primeiro mini-teste (MAT1) [S/N]: ")

if s1=="S":

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

while m1>20 or m1<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m1=float(input("Nota do primeiro mini-teste (MAT1): "))

else:

m1=" "

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

while s2!="S" and s2!="N":

print("\nPor favor responda sim(S) ou não(N)")

s2=input("Conhece a nota do segundo mini-teste (MAT1) [S/N]: ")

if s2=="S":

m2=float(input("Nota do segundo mini-teste (MAT1): "))

while m2>20 or m2<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m2=float(input("Nota do segundo mini-teste (MAT1): "))

else:

m2=" "

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

while s3!="S" and s3!="N":

print("\nPor favor responda sim(S) ou não(N)")

s3=input("Conhece a nota do terceiro mini-teste (MAT1) [S/N]: ")

if s3=="S":

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

while m3>20 or m3<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

m3=float(input("Nota do terceiro mini-teste (MAT1): "))

else:

m3=" "

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

while s4!="S" and s4!="N":

print("\nPor favor responda sim(S) ou não(N)")

s4=input("Conhece a nota do teste intermédio (MAT1) [S/N]: ")

if s4=="S":

TI=float(input("Nota do teste intermédio (MAT1): "))

while TI>20 or TI<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TI=float(input("Nota do teste intermédio (MAT1): "))

else:

TI=" "

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

while s5!="S" and s5!="N":

print("\nPor favor responda sim(S) ou não(N)")

s5=input("Conhece a nota do teste final (MAT1) [S/N]: ")

if s5=="S":

TF=float(input("Nota do teste final (MAT1): "))

while TF>20 or TF<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

TF=float(input("Nota do teste final (MAT1): "))

else:

TF=" "

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

while s6!="S" and s6!="N":

print("\nPor favor responda sim(S) ou não(N)")

s6=input("Conhece a nota correspondente à participação em aula (MAT1) [S/N]: ")

if s6=="S":

v=float(input("Participação (MAT1): "))

while v>20 or v<0:

print("\nPor favor introduza valores compreendidos entre 0 e 20")

v=float(input("Participação (MAT1): "))

else:

v=" "

l1=datetime.date(2017,11,29)

l2=datetime.date(2017,1,17)

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[" ",TI,TF,v," "]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

print("\nNOTA: Dados insuficientes para ilustrar o gráfico")

print()

else:

tm=[m1,m2,m3]

TM=heapq.nlargest(2,tm)

MT=(TM[0]+TM[1])/2

def a (MT,TI,TF,v):

a=(MT\*0.15)+(TI\*0.40)+(TF\*0.40)+(v\*0.05)

return a

MAT1=[["Média mini","TI","TF","Participação","Final"],["15%","40%","40%","5%"," "],["Datas mini",l1,l2," "," "],[MT,TI,TF,v,round(a(MT,TI,TF,v))]]

for z in MAT1:

print("|",z[0]," "\*(10-len(str(z[0]))),"|",z[1]," "\*(10-len(str(z[1]))),"|",z[2]," "\*(10-len(str(z[2]))),"|",z[3]," "\*(12-len(str(z[3]))),"|",z[4]," "\*(5-len(str(z[4]))),"|",file=fich1)

x=["m1","m2","m3","MT","TI","TF","Partipação","Final"]

y=[m1,m2,m3,MT,TI,TF,v,round(a(MT,TI,TF,v),2)]

def plotgraph (x,y):

plt.xlabel("Elementos de avaliação")

plt.ylabel("Notas obtidas")

plt.title("Classificações obtidas em MAT1")

plt.bar(x,y)

plt.ylim(0,20)

plt.yticks(np.arange(0,20+1,2.0))

plt.show()

plotgraph(x,y)

print()

fich1.close()

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

else:

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(tm,f)

pickle.dump(x,f)

pickle.dump(y,f)

f.close()

with open('fich.pkl','rb') as f:

if m1==" " or m2==" " or m3==" " or v==" " or TI==" " or TF==" ":

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

else:

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

tmread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(tmread)

print()

print(xread)

print()

print(yread)

f.close()

fich1.close()

if all1=="S":

def lerAfixaFich(nomeFich):

fich1=open(nomeFich,"r")

texto=fich1.read()

texto

print(texto)

fich1.close()

return

nomeFich="GAU"

lerAfixaFich(nomeFich)

with open('fich.pkl','wb') as f:

pickle.dump(AP,f)

pickle.dump(NFC,f)

pickle.dump(IG,f)

pickle.dump(IE,f)

pickle.dump(MAT1,f)

pickle.dump(docentes,f)

pickle.dump(disciplinas,f)

pickle.dump(x,f)

pickle.dump(y,f)

with open('fich.pkl','rb') as f:

APread=pickle.load(f)

NFCread=pickle.load(f)

IGread=pickle.load(f)

IEread=pickle.load(f)

MAT1read=pickle.load(f)

docentesread=pickle.load(f)

disciplinasread=pickle.load(f)

xread=pickle.load(f)

yread=pickle.load(f)

print(APread)

print()

print(NFCread)

print()

print(IGread)

print()

print(IEread)

print()

print(MAT1read)

print()

print(docentesread)

print()

print(disciplinasread)

print()

print(xread)

print()

print(yread)