**Description**

Hi there, in this project of 50 assignments you will get to **solve assignments related to simple python, conditional statements, loop structures, functions and at the end there are assignments in which we have integrated all topics.** This course is specially designed for python but one can solve the assignments in other programming language as well.

In case you won't be able solve the assignment you can navigate to the **solution segment to find the solution of the assignment.** We have managed all the assignments and there solutions in a well managed way. You can contact us to access the word document of all the assignments and solution of all assignments is already available in shared github repository.

Good luck

Distribution of assignments:

Assignment 1 – Assignment 5 -> **Level 1, Basic python**

Assignment 6 – Assignment 12 -> **Level 2, Conditional statements**

Assignment 13 – Assignment 21 -> **Level 3, For loops**

Assignment 22 – Assignment 24 -> **Level 4, While loops**

Assignment 27 – Assignment -> **Level 5, Functions**

Assignments 41 – Assignment 50 -> **Level 6, everything integrated**

Take your time and understand all the assignments. Once you are done with each assignment of this course,  you will hopefully be able to start your professional journey or journey of data science toolbox. Furthermore you will be able to make professional menu tasks. If you are willing for that, I can provide you a task a school management system task which will be very wide and consist of many options. We made that project on the requirement if Malaysian client. We can also provide you with code of that project.

**SECTION 1:**

**Instructions**

This section will contain python basic assignments which do not involve loop, conditional statements and likewise things. These will be tricky tasks and if you will be able to solve this without online help, you should be then proud on yourself. Otherwise, you can improve you understanding by taking online help and then feel proud that you now know how solve such tasks!

Objectives of this particular section:

* becoming familiar with the concept of storing and working with different data types in Python;
* Experimenting with Python code.
* becoming familiar with the concept of, and working with, variables;
* performing basic computations and conversions;

**1**

**Once upon a time in England**, John had three apples, Mary had five apples, and Adam had six apples. They were all very happy and lived for a long time. End of story.

Your task is to:

* create the variables: john, mary, and adam;
* Assign values to the variables. The values must be equal to the numbers of fruit possessed by John, Mary, and Adam respectively;
* having stored the numbers in the variables, print the variables on one line, and separate each of them with a comma;
* Now create a new variable named total\_apples equal to addition of the total apples.
* print the value stored in total\_apples to the console

**2**

Miles and kilometers are units of length or distance.

Bearing in mind that 1 mile is equal to approximately 1.61 kilometers, **complete the program in the editor so that it converts:**

* **miles to kilometers;**
* **kilometers to miles.**

You have to complete this program:

kilometers = 12.25

miles = 7.38

miles\_to\_kilometers = ###

kilometers\_to\_miles = ###

print(miles, "miles is", round(miles\_to\_kilometers, 2), "kilometers")

print(kilometers, "kilometers is", round(kilometers\_to\_miles, 2), "miles")

Do not change anything in the existing code. Write your code in the places indicated by ###.

In upper code we have decided 12.25 km and 7.38 miles. You can do conversion completing the program.

Run the program and see what happens

**3**

# input a float value for variable a here

# input a float value for variable b here

# output the result of addition here

# output the result of subtraction here

# output the result of multiplication here

# output the result of division here

print("\nThat's all, users!")

Your task is to complete the code in order to evaluate the results of four basic arithmetic operations.

The results have to be printed to the console. Write your code in place of #  according to the given instructions.

**4**

**Temperature converter**

* Write a Python program that converts temperature from Celsius to Fahrenheit. Your program should:
* Prompt the user to input a temperature in Celsius.
* Calculate the equivalent temperature in Fahrenheit using the formula: 𝐹=9/5 × 𝐶 + 32.
* Print the result.

Don't hesitate to use as many brackets in calculation as you want.

**5**

Doctor Ahmad needs a program in which patient write his weight and height and program should return BMI value of that patient.

Your strategy should be:

* Ask the user to input their weight in kilograms.
* Ask the user to input their height in meters.
* Calculate the BMI using the formula: 𝐵𝑀𝐼 = 𝑤𝑒𝑖𝑔ℎ𝑡 / ℎ𝑒𝑖𝑔ℎ𝑡\*\*2​. # (height is squared in formula)
* Output the BMI value to the console.

Using logical statements for inputs and final outputs will give clarity

**SECTION 2:**

**6**

Write a Python program that prompts the user to input a score and then prints out "you are passed" if the user has scored more marks than 40.

These are known as **if statements.**

ARE YOU DONE WITH THIS?

Now, update you program, it should be able to deal with other scenario as well. If the user enter numbers less than 40, the program should print out: "You are failed, better luck next time."

These type of conditional statements are known are **if else statements.**

Now gear up for the coming assignments..

**7**

**Grading system**

Write a Python program that prompts the user to input a score and then prints out the corresponding grade according to the following criteria:

* Score >= 90: "A"
* 80 <= Score < 90: "B"
* 70 <= Score < 80: "C"
* 60 <= Score < 70: "D"
* Score < 60: "F"

Add two further conditions in your program such that if user input a string instead on integer score, our program should denote that **"string is not valid score, kindly input a integer number next time."**. When should you place this statement? Explore that in your compiler.

* If input is "answer", then output should be: answer is not valid score, kindly input a integer number next time.
* If scores are less than 0 and greater than 100 then program should tell the user that **"marks are out of range. You score should be in between 0 and 100."**

Such statements are known as **if else if or else if statements.**

**8**

### Largest among Three Numbers:

Write a Python program that asks the user to input three numbers and then prints out the largest among them.

**9**

**Swapping values**

Write a program that ask user to input two numbers and store the values in **x** and **y.**

Your program should then swap the values such that x will get value of y and y will get value of x.

Unfortunately you can't create a third variable for this task.

sample:

enter a number: 5 (store value in x)

enter another number: 6 (store value in y)

Values swapped!

x = 6

y = 5

This task can take your time. You better understanding take out your notebook and imagine different scenarios. Furthermore you specifically don't need to use conditional statements in this task.

Give time to this assignment and properly understand your code.

**10**

**Triangle type checker:**

Create a Python program that determines the type of triangle based on its sides. Your program should:

* Prompt the user to input the lengths of three sides of a triangle.
* Use conditional statements to check if the triangle is equilateral, isosceles, or scalene.
* Print the type of triangle.

Equilateral triangle: Three sides are of same length.

Isosceles triangle: Two sides are of same length.

Scalene triangle:  All three sides are of different lengths.

You can use as many conditional statements as you want.

**11**

**Tax calculator**

In a City name Jabinga in molangala have heavy taxes on monthly income of each of there employee.

There rules are:

* 0% tax on income up to $300
* 10% tax on income from $400 to $700
* 20% tax on income from $700 to $1000
* 30% tax on income above $1000

Ten thousand of the total population was assigned just to calculate tax of each individual. Still sometimes few people used to claim that the calculations are wrong.

You need to create a program that takes income as a input from user and display the amount tax that this particular person has to pay.

Let me break down steps for you:

1- User input

2- Tax calculations

3- Output

4- Error handling  (if the user enters wrong input, we should tell him that input is not correct)

It is something interesting. Good luck.

Note: no such states exist in the world.

**12**

**Factors**

As a math teacher, you need to write a program for your students in which your student will input a number and program will indicate whether it is a multiple of 2 or 3 or 4 or 5 or 6 or 7 or 8 or 9 or 10.  In short it will give all factors of that number from 1 to 10.

Students can give any number. For that number your program should return all factors from 1 to 10.

For example if the user enters 20, our program should print all numbers that are factor of 20. In this case it will print: 2, 4, 5, 10.

If the user enter 15, our program should print: 3 and 5.

**SECTION 3:**

**13**

Now we are going to grasp for loop logics from basics in detail.  
Starting from very simple,

1. Write a code that prints numbers from 1 to 100 using for loop.
2. Write a code that prints '**Your name** is professional in python' 100 times,  (Replace 'your name' with your actual name)

These will be two different codes.

**14**

Let's get some advanced!  
Let say a class a school contain 20 students. Using for loop, write a code that ask user to enter name of student 20 times. Every time user enters a name, you will store that name in a list named 'students'.

**Note: Create a empty list before entering in the loop body.**

Do one more thing! print list in every iteration of loop for better understanding how elements are being added in the list.

To add elements in a list: *list.append(element)*

**15**

Let's calculate average marks of a whole class using loop structure. Let say class consist of 20 students. Your for loop should run 20 times and ask user to enter marks of student. Store all marks in a list.

Once data of 20 students in entered, you have list containing marks of all 20 students. Calculate average of class by this formula:    Sum of numbers / Total number of students

*Length of list will represent number of students*

You can use built function sum(list) and len(list) to calculate the average value.

**16**

Let's now merge conditional statements and for loops. Your task is to write a program that runs for loop 10 times. Each time it should:

* Take input from user
* Apply if else block to predict whether it is a even number or odd number
* If it is a even number, print 'Number is even!' and if number is odd, print 'Number is odd!'

**17**

li = [34, 54, 23, 05, 45, 65]

Your task is to create a for loop that calculate sum of all numbers of this list.

With every iteration of loop, print your current sum to see what's happening!

**18**

**Multiplication table**Your program should first take input number from user and output table of that number up to 10.

If input number is 3, output should look like:

3 X 1 = 3

3 X 2 = 6

3 X 3 = 9

3 X 4 = 12

3 X 5 = 15

3 X 6 = 18

3 X 7 = 21

3 X 8 = 24

3 X 9 = 27

3 X 10 = 30

**19**

**Students info**

Create two empty lists: student\_name and student\_marks.

Write a for loop in which user will give name and marks of students as input and you will store name in student\_name list and marks in students\_marks list. Enter data for 10 students.

Write another for loop. This for loop will print each student name with their marks in a logical way.

Sample output will be:

Ali got 45 marks

Ahmad got 60 marks

Mujtaba got 46 marks

Rehman got 60 marks

Murtaza got 34 marks

Abdullah got 40 marks

Hufaiza got 90 marks

Muneeb got 50 marks

Fahad got 56 marks

Zain got 79 marks

Names and marks of students are changeable. You will give these inputs during execution of first for loop.

**20**

When I was in grade 11, we had to find factorial of a number using paper and pencil.

I was fun for smaller number for larger numbers, it was lengthy and boring and sometimes calculations were wrong. Lets now create a program that will calculate factorial of a given number.

**What is factorial?**

Factorial of 3 = 3 X 2 X 1 = 6

Factorial of 4 = 4 X 3 X 2 X 1 = 24

Similarly any number multiplied by all numbers smaller than that number gives factorial value.

For example if number is 5, then 5 will get multiplied by all numbers smaller than 5 i.e. 1,2,3,4

Factorial of 5 = 5 X 4 X 3 X 2 X 1 = 120

Let's make a python code that will just take number as input and print its output.

Let's make this program together!

product = 1

# initially factorial is 1

fact = \_\_\_\_('Enter a number you want factorial of: '))

# Take proper input from user in integers.

# Initialize a for loop. Range of i should cover all number less than input but it should not include 0

# because 0 is not included in calculation of factorial. if input is 3, range should be: 1, 2, 3. Range

# should include input itself as well.

for i in range(\_, fact+\_):

# with every iteration, multiply value of product with current value of i and rewrite value of product

    product = product \* \_\_

print('Factorial of given number is: ' + str(product))

**21**

### Emergeny - Break statement

break keyword in loops is used to stop the execution. Now we will make a program that will run 100 times. But under certain conditions its execution will stop immediately. Let say you are a doctor. Whenever a patient comes, one of your worker enter disease name of that particular patient.

You want your system to stop taking input once the Heart patient comes. Whenever the disease name is 'heart' then the program should print 'Emergency situation Emergency situation' and no more inputs should be taken.

Write a for loop that will run 100 times. Under the body of loop, take input from user about disease name. Use a if statement to check whether the disease name is 'heart'. if it is, then print particular statement and break the loop at that point.

**SECTION 4:**

**22**

**While loops**

Let's start with while loops in python. Starting from making a simple while loop,

* Write a while loop that print numbers from 1 to 20.
* Write a while that print multiple of 3 from 30 to 90.
* Write a while loop that print your name one hundred times.

Output for third progam should look like:

1 - Ali

2 - Ali

3 - Ali

.

.

.

.

100 - Ali

**23**

**Guess game – atomic power**

Now let's create a simple guess game using loop.

Every guess game have one particular question and its answer. Our question will be:

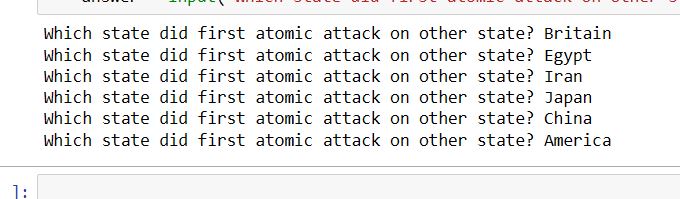
'Which state did first atomic attack on other state?' and answer to this question is America.

Write a while loop that will ask user the question  *'Which state did first atomic attack on other state?'*Again and again until the user enter America. Once we get the right answer, we will out of while loop.

You will need to create 'answer' variable with empty strings before entering while loop. Apply the condition such that loop will run until answer (variable) is not equal to 'America'.

Within the body of while loop, take input from user and store the value in variable 'answer'

**Sample output:**



**24**

Professor Amjad need a program for his math students. Students are aged  7-13. Professor want to make a program that will ask the student simple multiplication answer and keep asking the question until correct answer is entered. To generate two random numbers, we will use random module.

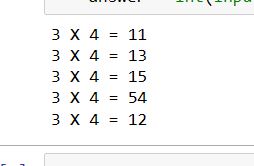
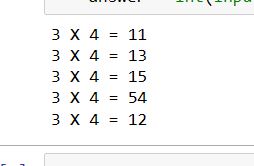
import random

a = random.randint(1,10)

These lines can generate  a random number from 1 to 10.

**Step by step guide**

* You will write a program which will generate two random numbers. Store your random number in variable *'num1'* and *'num2'.*
* Create a third variab



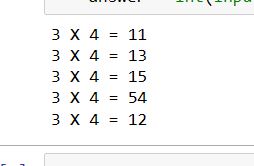
le *'product'* equal to product of two random number that we generated previously.

* Create variable *'answer'* .  This variable will be empty string as we did in previous assignments.
* Finally build a while loop that will run until answer is not equal to *'product'.* Within the while loop ask your question from the user and store answer in variable *'answer'.*
* Question is: num1 X num2  = ?

(num1 and num2 will be randomly generated numbers.)

You will have to do datatype conversion for this task for the input statement.

Sample output:



**25**

Let's make previous assignment more practical. Let say professor want us to make a program that ask the student questions number of times. For example it will repeat the assignment number of times. Let say he want us to ask 15 questions from student.

How can implement this? You can copy the code the paste it again and again and hence the program will ask questions again and again.. Right?

Well, try not to do this way! Use a for loop to repeat the process 15 times.

Hint: Make a for loop that will run 15 times and place your whole code within body of that for loop.

**26**

You may get bored of while loops, let's create something different

There are three seating categories at a stadium.

Class A seats cost 2000 rupees,

Class B seats cost 1500 rupees,

Class C seats cost 1000 rupees,

Write a program that asks how many tickets for each class of seats were sold, then displays the amount of income generated from ticket sales.

You may implement this without using while loop.

**27**

create a function 'profile' with two parameters: name and age. The function should print:

'Murtaza is 35 years old.' (Murtaza and 35 are arguments that we will pass when calling the function)

# complete function by filling the blanks

def profile(--,--):

print(---)

# check the function by calling it

**28**

Let suppose you have a particular program:

a = input('Write your first name: ')

b = input('Write your last name: ')

​

print('\nThanks for your cooperation')

Write your first name: harman

Write your last name: waheed

Thanks for your cooperation

You are writing a big program and this part of code repeat again and again. For saving time and instead of writing these three statements again and again, You can convert this into a function

Define a function 'into' with no parameter. And place the upper code in body of the function. Make any changes if needed and check the program

**29**

Lets now create a calculate() function which will take two numbers as parameters. These parameters will be 'num1' and 'num2'

This function will print result of addition, subtraction, division and multiplication of those two numbers

# complete the code

def calculate(--, --):

print('Addition of given numbers:', ---)

print('\nSubtraction of second number from first:', ---)

print('\nMultiplication:', ---)

print('\nFirst number divided by second:', ---)

# call your function with values of num1 and num2 as 2 and 2 to analyze the results.

**30**

We have created a function is 27th assignment. Let say all persons are of age 40 in that assignment. Very few are those who are not 40.

Make some changes in the function such that, it will automatically read age as 40 always when the program is called.

Also when in some cases we want to change the age, our program should handle those situations by passing the age value as first argument when calling the function

Hint: no hint

**31**

def func(a, b, c, d):

print(a)

print(b)

print(c)

print(d)

Call this function in two ways:

1 - give positional arguments

2 - give named arguments

When giving named argument also try to distort the sequence of parameters, for example give value of b first, then d, then c, then a

**32**

**Debug** the code the write a correct version below.

def range(100):

for i in range(10):

print()

**Functionality should be:** this function will print all numbers smaller than a given number (number should be parameter and later on we will assign its value when calling the function)

**33**

When we define the parameters, predefined parameter are always placed on end

Write a function 'players' with two predefined parameters and one other parameter.

-----------**Parameters**-----------

name

suports = 'cricket'

standard = 'great'

def ---(---,---,---):

print()

Update the print statement such that output when calling the function should look like:

**Pat cummins is a great cricket player.**

note that 'Pat cummins', 'great' and 'cricket' are arguments that we will pass to the function.

**34**

After you are done with assignment 33 function, lets call the function you created there with different techniques.

Which way will you run functions when outputs should look like:

**1 - Messi is a worst cricket player**

**2 - Messi is a astonishing football player**

**3 - Babar azam is a unpredictable cricket player**

**4 - Akmal shehzad dogar is a strong kabaddi player**

**5 - Azam khan is a active cricket player**

Give arguments when required. You can use named arguments as well positional arguments

35

Write a function 'motivate' with only one parameter 'iter\_'. We will be able to change its value but by default it should be 10

What function will do: it will print 'I can do it' as many times as specified in the parameter.

36

print('--------------')

print('| |')

print('| |')

print('| |')

print('| |')

print('--------------')

these statments are used to draw a square.

Instead of of using these 7 lines again and again to draw a square, lets create a function 'sq' which will output exactly same square when we will call that function

**37**

Lets's create some simple functions. write a function that will take one number as parameter and print whether the number is even or odd after evaluating this through conditional statements

**38**

Rewrite following functions error free

hello()

def hello(name='Ali Abbas', age= 40, occupation ='Wakeel'):

print('Ali is a 40 years old wakeel')

def code(age=40, name, occupation):

print(name, 'is a', age, 'years old', occupation)

def movies(name='Lord of rings', part = 3):

print('Movie', name, 'has' + part, 'parts')

def person(address, contact, name, organization, name)

print('thanks for data')

**39**

import random

for i in range(4):

num1 = random.randint(1,10)

num2 = random.randint(1,10)

answer = ''

while answer != num1\*num2:

answer = int(input(str(num1) + ' X ' + str(num2) + ' = '))

Remember this code? This is solution of assignment number 25. See this code and read of instructions if assignment 25. Once you get this code,

Lets make a function of this code.

function name: 'mul'.

Parameter: num

In third line the integer within range() will be replaced by num so that it will ask as many questions as you will specify while calling function. Define and call your function to see the results.

**40**

Write a simple function that will take one number as parameter and print its multiples from 1 to 10 in logical way.

For example if you give number 9, output should looks like:

9 X 1 = 9

9 X 2 = 18

9 X 3 = 27

.

.

.

9 X 10 = 90

**41**

Let’s write a menu for school system. Functionality of our program will be:

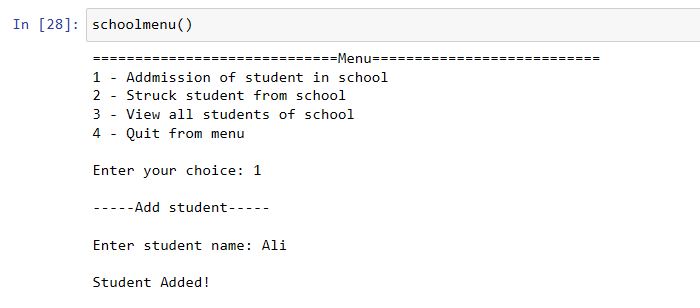
1 - Admit students in school

2 - Remove students from school

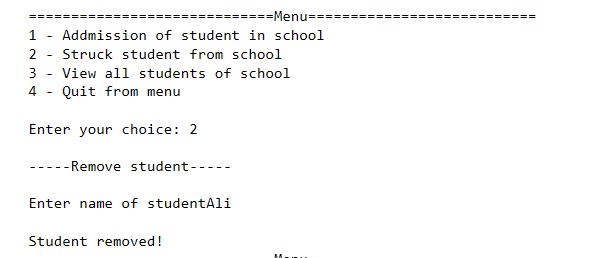
3 - View all students of school

Here are some images of output of program for your guidance.

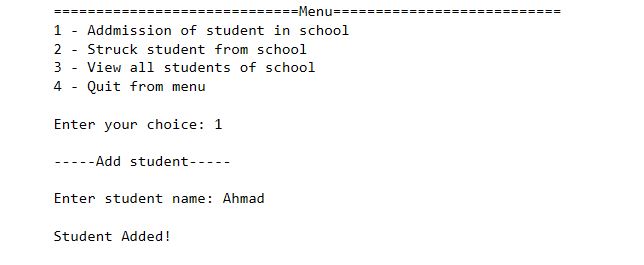
**Admission of student**



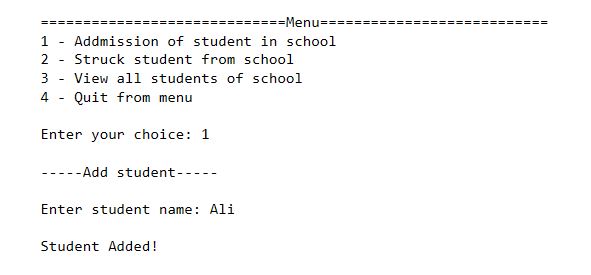
**Removal of student**



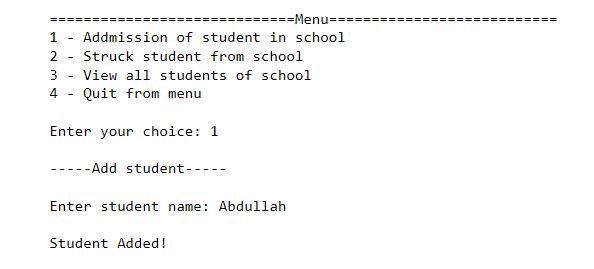
**Admission of student**



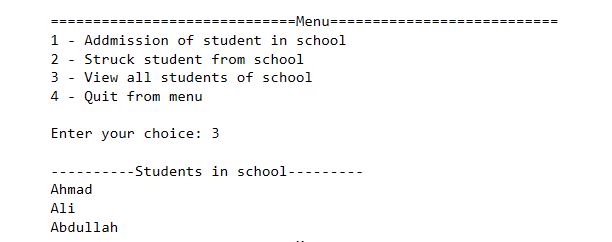
**Admission of student**



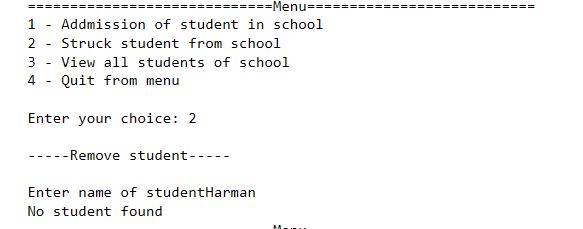
**Admission of student**



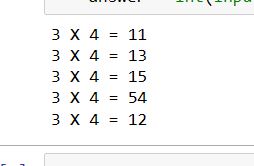
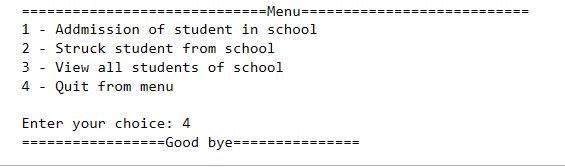
**View students**



**Removal of student who is not present in school**



**Exit program**



Furthermore here are some instructions for this project:

1 - Use list to store data (student names).

2 - For task 1 use list built-in method to add student in list

3 - For task  2 use conditional statement to check whether the student is present in school or not. Remove student in he is in school by using list built-in method 'remove()'

4 - For task 3 use for loop and iterate over the list to to display all students

5 - For implementation of last point, you need to run program again and again unless user decide to exit. For that enclose whole program in while loop such that it should run again and again. Use conditional statement to check whether user want to exit, then you may use break statement to stop while loop.

This is how whole execution of upper images looks like:



For our case, we implemented this in a function but you can make it without function as well.

**42**

Our purpose is to

* hold data of patients. We will store name and age of each of patient.
* our hospital can only accommodate 40 patients. So if patients are more than 40, we will indicate that we cannot accommodate this much patients.
* Our hospital do not provide services to old age persons. If there is any patient more than 60 years old, we will not add him in hospital data.

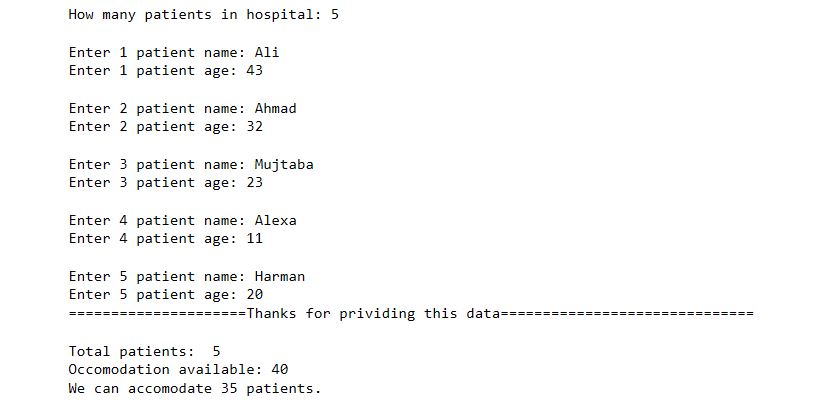
Instructions:

* Take input from user about number of patients, then iterate for loop as many times as there are patients. Each time For loop run, store data of patient name and age in a list and then append that list in a major list that will at the end contain data of all the patients.
* Use conditional structure to check if age is greater than 60. If it is greater, we won't admit that patient in hospital. It means we will not store data of the patient.
* Now print how many more patients we can accommodate in our hospital.

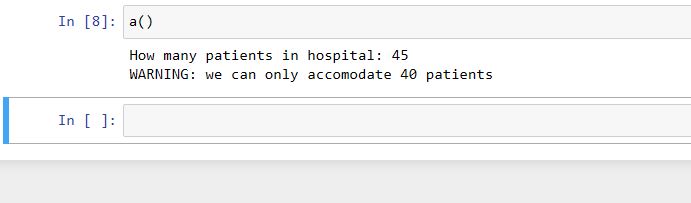
In case user input number of patients as more than 40, you will have to indicate that we cannot accommodate this number, and program should stop further execution.

Here are some sample executions but obviously your statements can be different from mine.

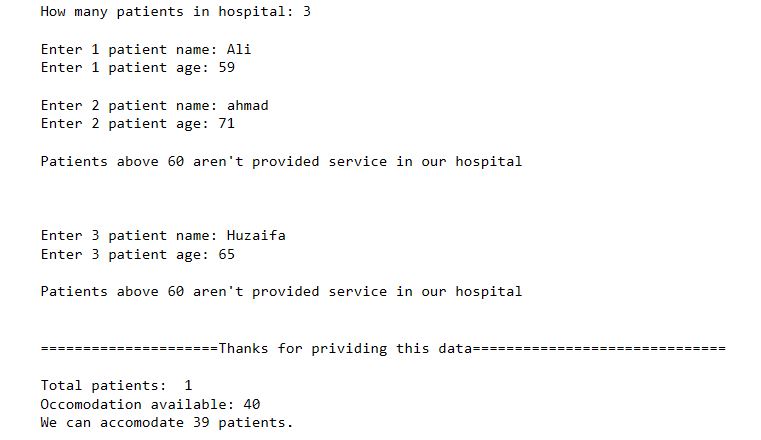
**Case 1: Input data smoothly**



**Case 2: Input data of more than 40 patients**



**Case 3: Input data of patients of age more than 60 years**



**Good luck!**

**43**

An organization need you to calculates weekly employee payroll for all employees.

For each employee, ask the user for the employee ID number, the hourly wage rate, and the number of hours worked during a week.

Each employee is paid 1.5 times their regular hourly rate for all hours over 40. A tax amount of 3.625 percent of salary is deducted.

At the end print total salary of all employees and number of employees.

INSTRUCTIONS:

- Create two variables with value 0 to store total salary and total number of employees so that we can print that data at the end

- Initialize a while loop, take ID, hourly\_wage, hours\_worked as inputs and do all the computations to calculate net salary of that employee. This while loop should stop running when user enter 0 as employee id

- Also write code to add salary of employee and employee number in two variable that we created in the first step.

- Once we are out of while loop, program should indicate total salary of all employee and total number of employee.

**44**

Let's write something functional for someone's budget plan

**Its functionality should be:**

* ask user for his expenses continuously unless user enters 0
* print total expenses
* print whether the person is in Surplus, Deficit, or Balanced.

Surplus: if monthly income is more than expenses

Deficit: if monthly income is less than expenses

Balanced: if monthly income is equal to expenses

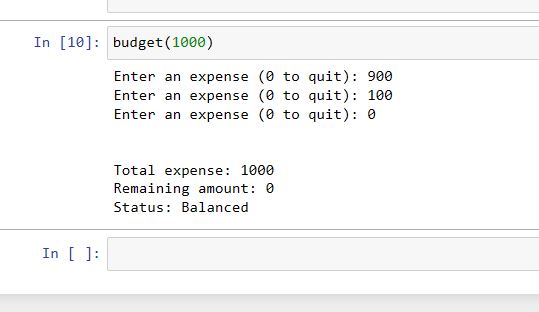
* How will you know the monthly income of person?

**Instructions**

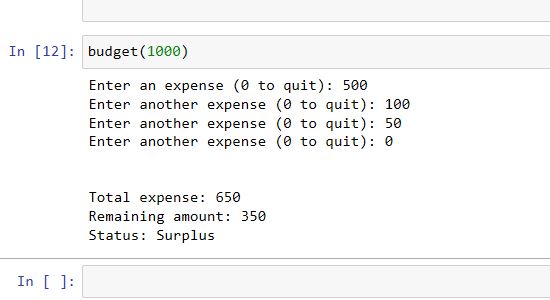
* Create a function 'budget' with a parameter 'available' such that parameter will indicate the total budget of a month.
* Write your all code in function and test that.
* At the end it should print: total monthly budget, remaining balance, status of person (surplus, deficit, or balanced)

**Sample outputs:**

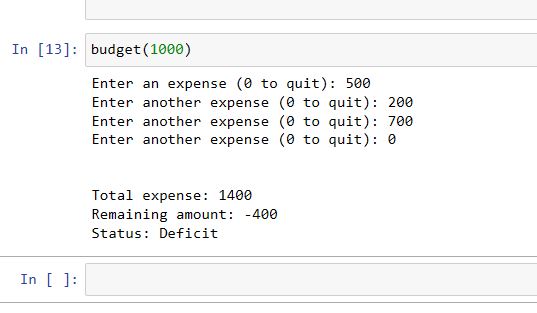
**Case 1:** When expenses are balanced:



**Case 2:** When monthly income is more than expenses



**Case 3:** When monthly income is less than expenses



**45**

**Menu programming**

Creating a menu functional menu for restaurant will be a good practice for your logics now,

This exercise will only include function and conditional statement.

Here is some part of code:

def res\_menu():

print('=================MENU===============')

print('1 - Burger beef - 600')

print('2 - Tortilla Wraps - 800')

print('3 - Chai Paratha - 200')

print('4 - Chai - 150')

print('5 - Shake (Mango, banana, date, chiku) - 600')

menu = [600, 800, 200, 150, 600]

choice = int(input('\n\nEnter your option: '))

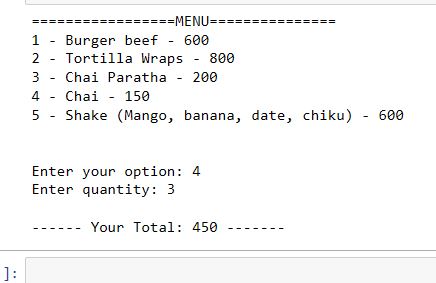
---

You will continue this menu to meet these requirements:

* In the last line of upper code, user will enter his choice about what he want to order from the menu, extract prize of that item from the list 'menu' using list indexing.
* Ask user for quantity of item he want to order and then print out total bill

Once done,  run your code and analyze the outputs.

**Sample run:**

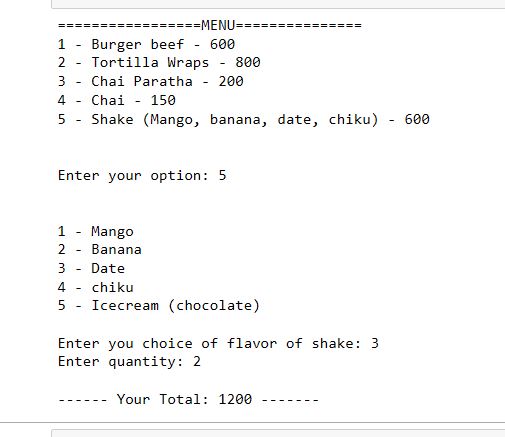


**After you are done with this, follow these statements**

As you see in the menu 5th option, there are shake of different flavors. Lets make refine our code such that if user want to order the shakes, we will ask him to enter the flavor as well.

This will not change the bill for user but this will make our menu more logical and detailed. You can use conditional statements for this implementation.

**Sample run:**



**46**

Let us refine the code of previous assignment (45)

Previously we were taking only one item in order, but customer may want to make his order of more than one item. For example he may want to order shakes as well chai paratha and maybe more.

Do something such that program will take order from user again and again until user press zero to stop the order. Then represent total bill.

**47**

**Money balance for meals**

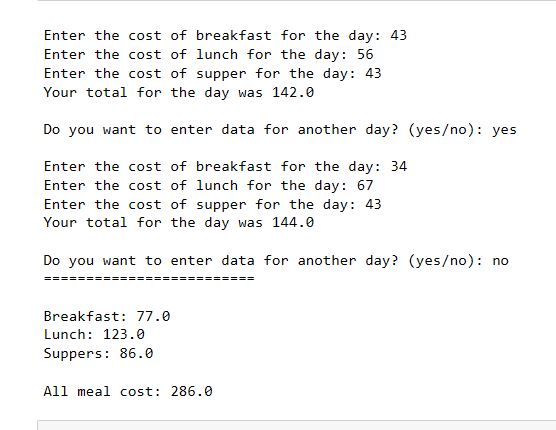
write a function that asks a user the costs of breakfast, lunch, and supper. Print total cost of that day after these inputs.

After entering data for each day asks the user whether they want to enter data for another day.

At the end:

Print cost of all the breakfast, all the lunch, and all the supper separately. Also print Total cost of all meals. For this purpose you may need to create few variables in the beginning of function to hold data of all meal costs.

**Your statements can be different, but here is a sample output:**



**48**

Let's plot some graphs. But we didn't covered graphs yet..No worries we will use the logics that we already know to make some graphs.

**Our task is to**

visualize sales in a market through '\*'

**How**

* Let suppose you own 5 shops in a market. Write a program such that user will input sales of those 5 shops of 1 day. You can do this manually but for the sake of practice, use for loop. Store all sales in some variable.
* If sale of a shop is 2500, there will be 25 '\*\*' in the graph. If there will be sale of 800, there will be 8 '\*' in the graph.

**This is how output should look like**

Enter today's sales for store 1: 1000 [Enter]

Enter today's sales for store 2: 1200 [Enter]

Enter today's sales for store 3: 1800 [Enter]

Enter today's sales for store 4: 800 [Enter]

Enter today's sales for store 5: 1900 [Enter]

SALES BAR CHART

Store 1: \*\*\*\*\*\*\*\*\*\*

Store 2: \*\*\*\*\*\*\*\*\*\*\*\*

Store 3: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Store 4: \*\*\*\*\*\*\*\*

Store 5: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* If there are 400 worth sales, you can calculate number of '\*' by dividing it by 100 -> 400 / 100 = 4
* Then you will be able print 4 '\*' (for this case) by using: '\*' \* 4
* Store data in proper variable.
* You will be able to generate these results without using loop structures and that will be fine as well.

Use output scenario we have given to better understand the task.

**49**

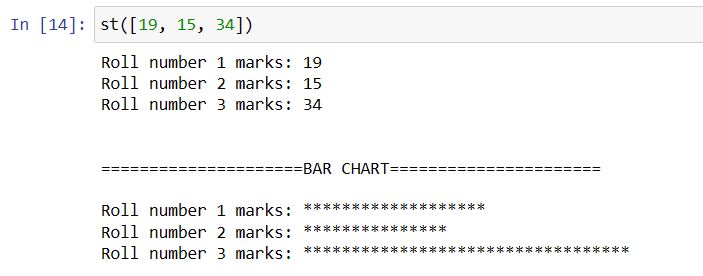
In this assignment, we will write a program that will take a list of marks of students in accordance with there roll numbers.

**[23, 43] this will indicate that roll number 1 marks are 23 and roll number 2 marks are 43**

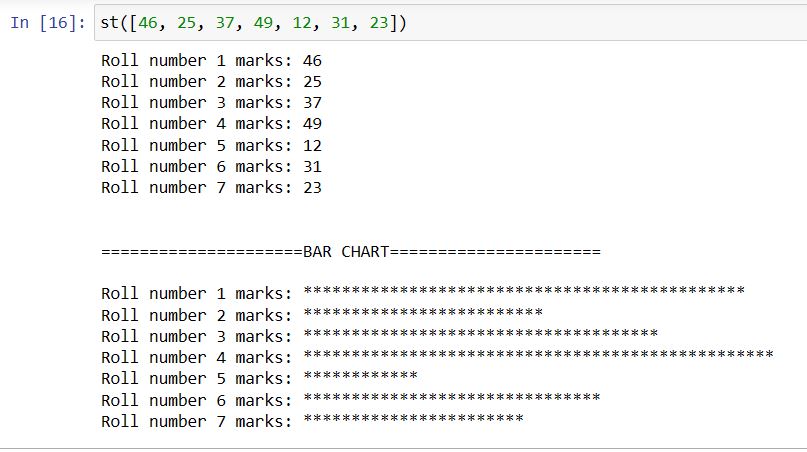
Marks are out of 50. You will have to make bar charts for each student in list.

* First print all marks of students with roll number using a for loop.
* After that print bar chart of student marks.

**Sample outputs:**



**Another**



**50**

Write a program that lets the user play the **game of Rock, Paper, Scissors** against the

computer. The program should work as follows.

1. When the program begins, create a list ['rock', 'paper', 'scissor'] and let the computer select one of these randomly.
2. The user enters his or her choice of “rock”, “paper”, or “scissors” at the keyboard. (You can use a menu if you prefer.)
3. The computer’s choice is displayed.
4. A winner is selected according to the following rules:

* If one player chooses rock and the other player chooses scissors, then rock

wins.

* If one player chooses scissors and the other player chooses paper, then scissors wins.
* If one player chooses paper and the other player chooses rock, then paper

wins.

* If both players make the same choice, it is a tie

**Use as many conditional statements as you want.**

Do all the implementations in function so that we can play game again and again.