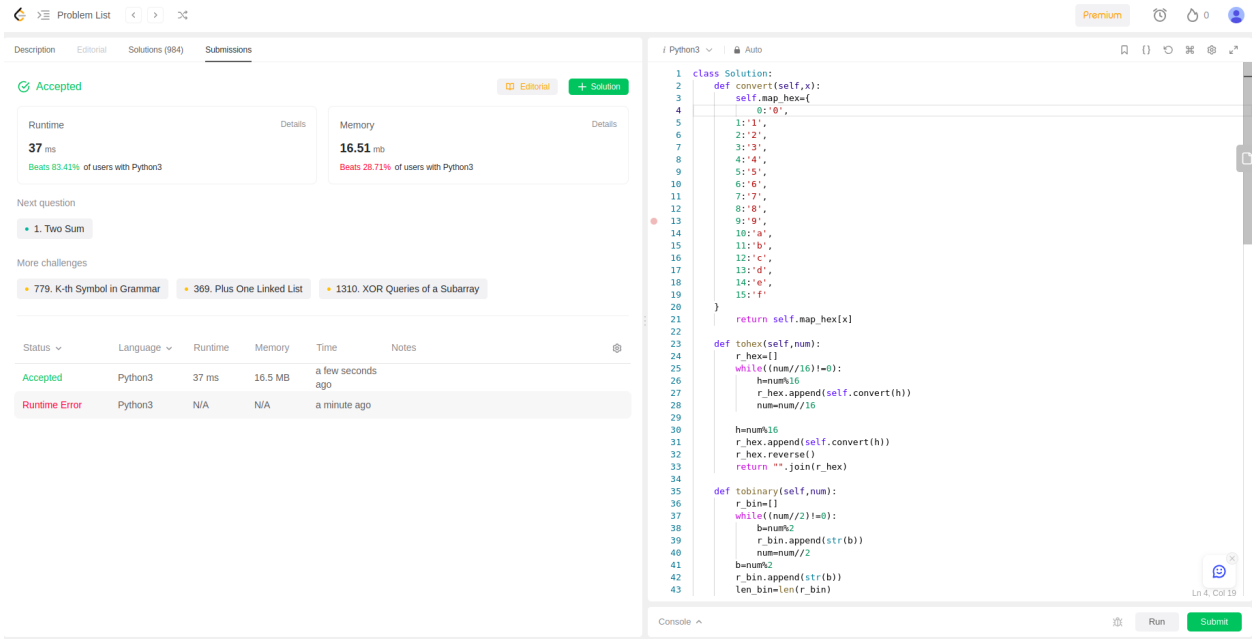
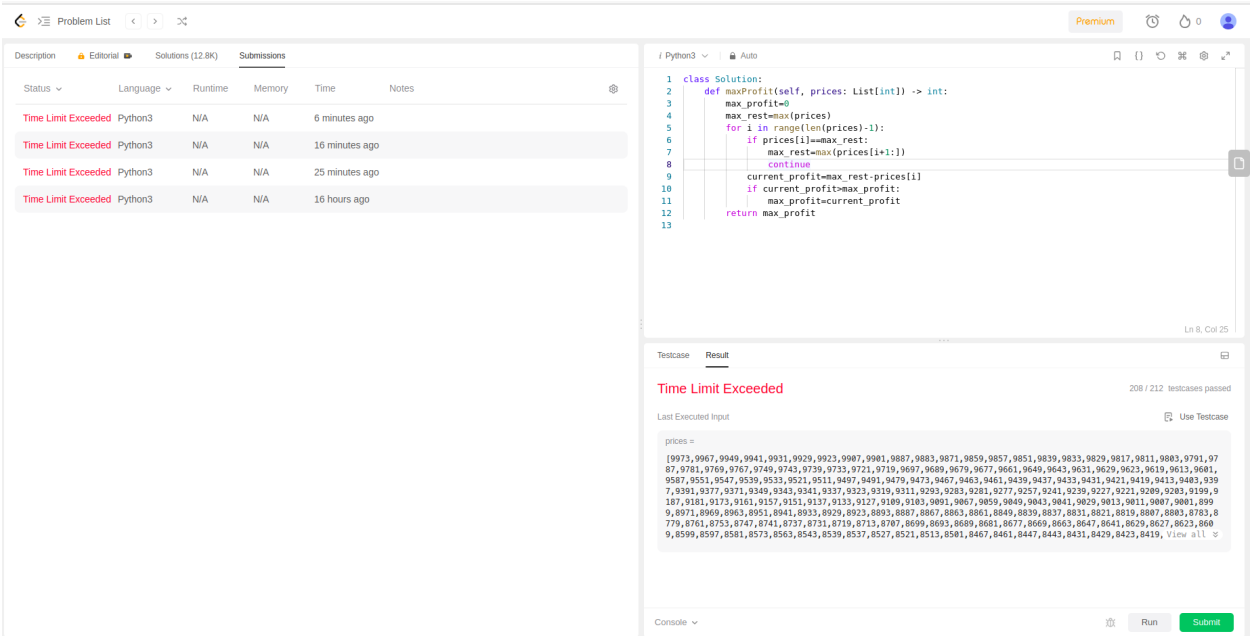
1. [Convert a Number to Hexadecimal]



There was two cases positive decimal integer and negative integer
For positive integer it convert the num directly to hexdecimal by divide the num over 16 and store the reminder for each operation
Then get the orders of the reminder from down to up to get the equivelent number in hexdecimal

For decimal negative number
1- first get the absolute value of the integer and convert it to binary representation
2- flip the binary number as it is a negative number
3- add the complement number with the binay number
4- convert the resulted binary number to hexdecimal number
5 - join the result and get the well-format string for it

2. [Best Time to Buy and Sell Stock]

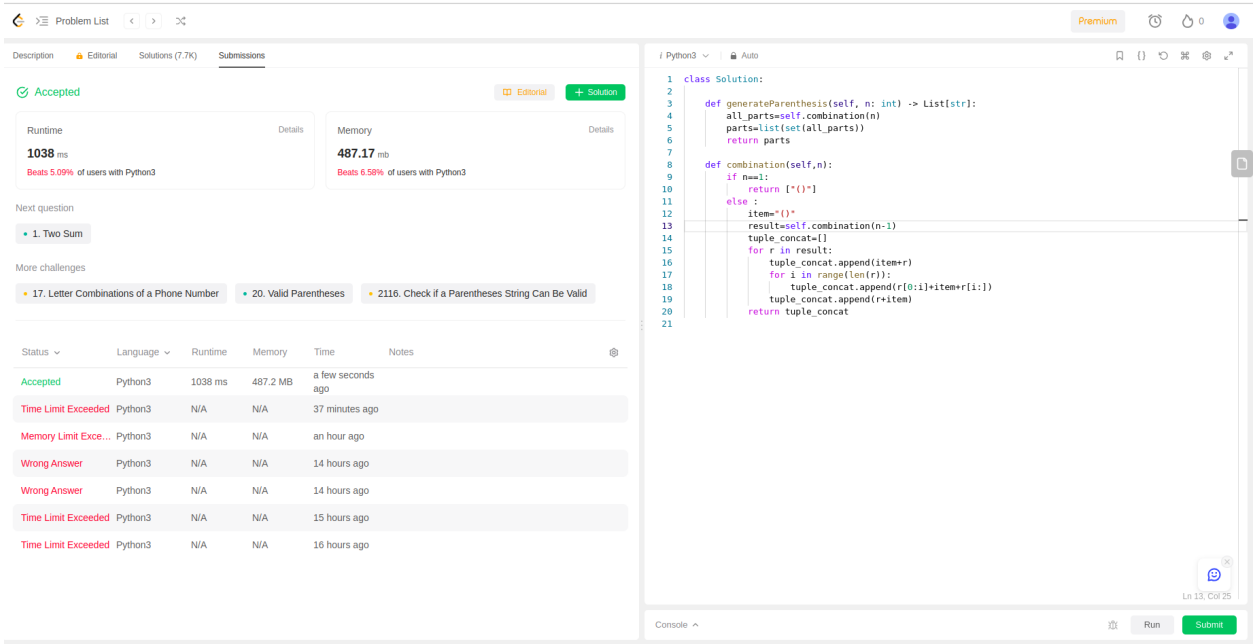


It get a Time Limit Exceeded

I think it should be in O(n)
And get the result in just single one loop

So we need for the single time that visit an item of the array
we also must calculate the current_profit that occurred using this item with the rest of the items of the array
Then compare the current_profit with the stored max_profit

3. [Generate Parentheses]:



It is a recursive problem with the base case “()”
Then for each recursive level it add the new added item “()” to each differnet available position
To construct all different permutation string

Q2:
what different types of model inheritance are used in odoo16?
odoo provides three types of inheritance
1- class inheritance (extension)

- it is used to add new fields and methods to existing models
at the database layer it doesn't create new database table for the inheriting class
and the added or chanched fields added and chanched to the same parent database model

fields is incrementally modified , mean if modeify and change attributes on the existing fields of the parent model
child method overwrite parent method if we didn't use super constructor to call superclass method the parent functionality will discard

it defined using _inherit which specify the model that will be extented by the inheriting model

2 - prototype inheritance
copy the entire defination of the existing model
it has it's own database table and indepentent from parent model
but copies all the fields , attributes and methods from the parent model

Since it still inherits from the parent class, any subsequent modifications to the parent class will also affect the inheriting model.
it like have a reference to the a parent class any modification to the parent will be existing in the child class also

it define using attribute of _name and _inherit where _name specify the name of the new inheriting model while _inherit it has the name of the parent class

3- delegation inheritance

it apply inheritance mechnism through relational fields

for example if have class model called member that inherit for patner mode through patner_id many2one relation
it actually link the instance of the member model with the corresponding instance of the partner model
so throught the member model can we access the fields of the patner model easily

It also supports polymorphic inheritance, where we inherit from two or more other models

It also works only for fields and not for methods so parent method can't be accessd using child method

it defined using _inherits which take a dictionay have the key it the parent class and the value is the relaional fields that link the two model together

3. Q3: Make a connection to a local database using psycopg2 and list all products in the system.

- Uploaded to the github repo

4. Q4: What is an external ID, and how can you use it in building views?
    is a string that refers to a record in the database
    The IDs themselves are records of the ir.model.data model.

    for examples :
    if we run the query select * from ir_model_data;
    from psql terminal
    it show that the table have the following fields
    1- id
    2- create_uid
    3- create_date
    4- write_date

5- write_uid
6- res_id
7- noupdate
8- name
9- module
10- model

where name field have the actually value of the string of the xml ID
how can you use it in building views?

- Search for a view
- View Inheritance

5. Q5: Given a model "smart_support" with a field "total_amount," how would you inherit the model and create a new field to calculate a 5% tax on "total_amount" using computed fields?

- Uploaded to github repo

6. Q6: Briefly explain Wizards, Reports, and Actions in Odoo 16.

- Wizards: where the records of the models.TransientModel are periodically cleaned up in the database.

these are used to create wizards or dialog
boxes, which are filled in the user interface by the users and are generally used to perform actions on the persistent records of the database.

- Reports:

Reports are written in HTML/QWeb You can use the usual Qweb flow tool control. The PDF rendering itself is performed by wkhtmltopdf
If you want to create a report on a certain model, you will need to define the report and the report template it will use.

- Actions :

in Odoo ERP actually defines the behavior of the system in response to user actions for example : a login done by the user, action button

Q9: Describe the composition (structure) of an Odoo Module in version 16

1- __manifest__.py file that used by odoo to detect addon module
2- odoo addon can have three type of files

- Python code is loaded by the __init__.py files
- Data files that are to be declared in the data and demo keys
- Web assets such as JavaScript code and libraries, and QWeb/HTML templates

The add-on files are to be organized into the following directories

➔ models/ contains the backend code files

- ➔ views/ contains the XML files for the user interface
- ➔ data/ contains other data files with the module's initial data
- ➔ demo/ contains data files with demonstration data
- ➔ i18n/ is where Odoo will look for the translation .pot and .po files
- ➔ security/ contains the data files that define access control lists such as
  - ◆ a ir.model.access.csv file
  - ◆ an XML file to define access groups
  - ◆ record rules for row-level security
  - ◆ controllers/ contains the code files for the website controllers
  - ◆ static/ is where all web assets are expected to be placed
  - ◆ wizard/ contains all of the files related to wizards
    - wizards are used to hold intermediate data
  - ◆ report/ : Odoo provides a feature to generate PDF documents such as sales orders and invoices.
    - This directory holds all the files related to PDF reports.

Any files that didn't defined in the __manifes__.py or in __init__.py will be ignored and won't be loaded

10. Q10: Write simple code to append a Menu Item called "Branches" to Odoo's user menu in version 16.

- Uploaded to github repo

12. Q12
Define method overloading and method overriding.?
Method overriding :
The child method and parent method have the same method name and the same signature but the implementation changed in each class and that what makes the polymorphism

Method overloading :
The child method and parent method have the smae method name but with different signature and implementation
Python didn't support method overloading

13. Q13:
What is inheritance in programming?
Is a mechanism that enable the child class to inherit fields , properties and methods from the parent class , overwrite the parent implementation for it's need

14. Q14:
Define a superclass in object-oriented programming.

Any Class or InterFace have the structure of methods and attribute that will be reused by the child class and implemented

15. Q15:
Mention three field types used in Odoo 16 models

- Computed Field defined by compute function that implement have to set the value for the computed field
- Related Field it like the computed filed but computed using the path traverse of the related filed
- Relational Field : reference for records of other models
    - Many2one
    - One2many
    - Many2many

16. Q16:
Highlight the differences between Lists and Tuples in Python.

List is mutable type so the values can changed
Tuple is immutable type so the values didn't change
It not allowed to assign values once created