Task 9 : Implement Exceptions and Exceptional handling

Aim :- To implement exceptions and Exceptional handling in Python.

Algorithm :-

1. start the program
2. Initializes a list of grades (eg., [85,90,78,92,88])
3. prompts the users to enter the index of the grade they wish to view.
4. Attempts to display the grade at the specified Index.
5. If the index is out of range, catches the IndexError and prints an error message, "Invalid index please enter a valid index."

Program :-

```
#Initialize the list of grades
grades = [85,90,78,92,88]
# Display the grades list
print("Grades List", grades)
# prompt the user to enter the index of the grade they want to view. try:
index = int(input("Enter the index of the grade you want to view:"))
    #Attempt to display the grade at the specified index.
    print(f"The grade at index{index}is:{grades[index]}")
except IndexError:
    # Handle the case where the index is out of range
    print("Invalid index. Please enter a valid index")
except ValueError
    # Handle the case where the input is not an integer
    Print("Invalid input. Please enter a numerical index")
```

Output :-

```
Grades list: [85, 90, 78, 92, 88]
Enter the index of the grade you want to
  view : 10
Invalid index. Please enter a valid index.
```

Output :-

Enter the numerator: 10
Enter the denominator: 0
ERROR !
Error: Division by zero is not allowed

Problem 9.2 :- You are developing a python calculator program that performs basic arithmetic operations one of the key functionalities is to divide two numbers entered by the user.

Algorithm :-

1. Starts the program
2. Prompts the user to enter two numbers: a numerator and a denominator.
3. Attempts to divide the numerator by the denominator
4. If the denominator is zero, catches the ZeroDivisionError and displays an error message "Error: Division by zero is not allowed".

Program :-

```
# Function to perform division
def divide_numbers():
    try:
        #prompt the user to enter the numerator
        numerator = float(input("Enter the numerator: "))
        #prompt the user to enter the denominator
        denominator = float(input("Enter the denominator: "))
        # Attempt to perform division
        result = numerator / denominator
        print(f"Result: {result}")
    except ZeroDivisionError:
        #Handle division by zero error
        print("Error: Division by zero is not allowed.")
    except ValueError:
        #Handle invalid input that is not a number
        print("Error: Please enter valid numbers!")
# Call the function to execute the division operation
divide_numbers()
```

Output :

```
Enter a number : 15
Exception occurred : Invalid Age
```

Problem 9.3 : You are building a python application to determine if a person is eligible to vote based on their age. According to the rules, only individuals who are 18 years or older are allowed to vote.

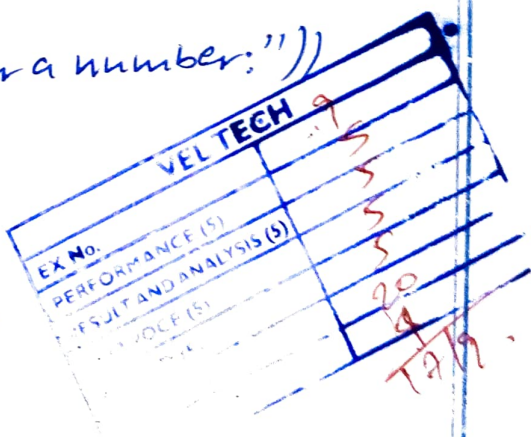
Algorithm :

1. Define the custom exception
2. prompt the user for input
3. check if the age is below 18
4. Raise an exception if the condition is met
5. Handle the exception with a custom error message

Program :

```
#define python user-defined exceptions
Class InvalidAgeException(Exception):
  "Raised when the input value is less than 18"
   pass
# you need to guess this number
number = 18

try:
   input_num = int(input("Enter a number:"))
   if input_num < number
      raise InvalidAgeException
   else:
      print("Eligible to vote")
   except InvalidAgeException:
      print("Exception occurred: Invalid Age")
```



Result : Thus, the program for implement Exceptions and Exceptional handling is executed and verified successfully.