In [ ]:
```python
# Q1. Write a program to handle the exception of ZeroDivisionError.
try:
    a=int(input("Enter 1st number"))
    b=int(input("Enter 2nd number"))
    c=a/b
    print(c)
except ZeroDivisionError as e:
    print(e)
    # print("It's an error")
```

```
division by zero
```

In [ ]:
```python
# Q2. Write a program to handle the exception of IndexError.
# [Handle the exceptions with ‘Try’, ‘Except’, ‘Finally’ and ‘Raise’ according to the requirements in the program.]

try:
    arr1=["koushik","das",1,2]
    print(arr1[4])
    print(arr1[3])
    a=int(input("Enter 1st number"))
    b=int(input("Enter 2nd number"))
    if(b==0):
        raise ZeroDivisionError("ZeroDivisionError")
    else:
        print(a/b)
except IndexError as e:
    print(e)
finally:
    print("The code ran successfully")
```

```
list index out of range
The code ran successfully
```

In [ ]:
```python
# Q1. Write a program using the Regular Exception and create a function that accepts a string and searches it for a valid phone number.
# Return the phone number if found.
# A valid phone number may be one of the following:
# (xxx)-xxx-xxxx
# xxx-xxx-xxxx

import re
def func1(text):
    pattern1=r'[(?][0-9]{3}[)?]\-[0-9]{3}\-[0-9]{4}'
    pattern2=r'[0-9]{3}-[0-9]{3}-[0-9]{4}'
    match1=re.search(pattern1,text)
    match2=re.search(pattern2,text)
    if match1 or match2:
        print("Valid phn number")
    else:
        print("Invalid phn number")
text=str(input("Enter the string"))
func1(text)
```

```
Invalid phn number
```

In [ ]:
```python
# Q2. Write a function that employs regular expressions to ensure the password given to the function is strong.
# A strong password is defined as follows:
# ·       at least eight characters long
# ·       contains one uppercase character
# ·       contains one lowercase character
# ·       has at least one digit
# ·       has at least one special character
# [For instance: Christ@123]

import re
def func1(text):
    pattern2=r'[A-Z]+'
    pattern3=r'[a-z]+'
    pattern4=r'[0-9]+'
    pattern5=r'[@ - / , .]+'
    match2=re.search(pattern2,text)
    match3=re.search(pattern3,text)
    match4=re.search(pattern4,text)
    match5=re.search(pattern5,text)
    if match2 and match3 and match4 and match5:
        print("Password Set Successfully")
    else:
```

```
        print("Set a Valid Password")
text=str(input("Enter the string"))
func1(text)
```

```
Password Set Successfully
```