1.Write a program that asks the user for a list index and prints the value at that index from a predefined list. Handle the IndexError and ValueError exceptions.

```
def access_list_element():
    my_list = [10, 20, 30, 40, 50]
    try:
        index = int(input("Enter the index: "))
        value = my_list[index]
        print("Value at index", index, "is:", value)
    except IndexError:
        print("IndexError: Index is out of range.")
    except ValueError:
        print("ValueError: Invalid input. Please enter an integer.")

access_list_element()
```

```
Enter the index: 2
Value at index 2 is: 30
```

```
def access_list_element():
    my_list = [10, 20, 30, 40, 50]
    try:
        index = int(input("Enter the index: "))
        value = my_list[index]
        print("Value at index", index, "is:", value)
    except IndexError:
        print("IndexError: Index is out of range.")
    except ValueError:
        print("ValueError: Invalid input. Please enter an integer.")

access_list_element()
```

```
Enter the index: 5
IndexError: Index is out of range.
```

```
def access_list_element():
    my_list = [10, 20, 30, 40, 50]
    try:
        index = int(input("Enter the index: "))
        value = my_list[index]
        print("Value at index", index, "is:", value)
    except IndexError:
        print("IndexError: Index is out of range.")
    except ValueError:
        print("ValueError: Invalid input. Please enter an integer.")

access_list_element()
```

```
Enter the index: keer
ValueError: Invalid input. Please enter an integer.
```

1. Create a program to validate exam scores entered by the user. Use a custom exception to handle invalid scores.

Create a custom exception class called InvalidScoreError that will be raised if:

The score is less than 0.

The score is greater than 100.

Write a function called validate_score(score) that:

Takes a score as input.

Raises InvalidScoreError if the score is not in the valid range (0–100).

Returns a success message if the score is valid.

Use try-except blocks to: Catch the custom exception and print an error message.

```
class InvalidScoreError(Exception):
    pass

def validate_score(score):
    if score < 0 or score > 100:
        raise InvalidScoreError("Invalid score. Score must be between
0 and 100.")
    else:
        return "Score is valid."

try:
    score = int(input("Enter the exam score: "))
    result = validate_score(score)
    print(result)
except InvalidScoreError as e:
    print(e)
except ValueError:
    print("Invalid input. Please enter an integer.")
```

```
Enter the exam score: 101
Invalid score. Score must be between 0 and 100.
```

```
class InvalidScoreError(Exception):
    pass

def validate_score(score):
    if score < 0 or score > 100:
        raise InvalidScoreError("Invalid score. Score must be between
0 and 100.")
    else:
        return "Score is valid."

try:
    score = int(input("Enter the exam score: "))
    result = validate_score(score)
    print(result)
except InvalidScoreError as e:
    print(e)
except ValueError:
    print("Invalid input. Please enter an integer.")
```

```
Enter the exam score: -2
Invalid score. Score must be between 0 and 100.
```

```
class InvalidScoreError(Exception):
    pass

def validate_score(score):
    if score < 0 or score > 100:
        raise InvalidScoreError("Invalid score. Score must be between 0 and 100.")
    else:
        return "Score is valid."

try:
    score = int(input("Enter the exam score: "))
    result = validate_score(score)
    print(result)
except InvalidScoreError as e:
    print(e)
except ValueError:
    print("Invalid input. Please enter an integer.")
```

```
Enter the exam score: 91
Score is valid.
```

```
class InvalidScoreError(Exception):
    pass

def validate_score(score):
    if score < 0 or score > 100:
        raise InvalidScoreError("Invalid score. Score must be between 0 and 100.")
    else:
        return "Score is valid."

try:
    score = int(input("Enter the exam score: "))
    result = validate_score(score)
    print(result)
except InvalidScoreError as e:
    print(e)
except ValueError:
    print("Invalid input. Please enter an integer.")
```

```
Enter the exam score: score
Invalid input. Please enter an integer.
```

3.You have a dictionary. Ask the user to enter a key and display the corresponding value. Handle the KeyError.

```
def access_dictionary_element():
    my_dict = {"apple": 1, "banana": 2, "orange": 3}
    try:
        key = input("Enter the key: ")
        value = my_dict[key]
        print("Value for key", key, "is:", value)
    except KeyError:
        print("KeyError: Key not found in the dictionary.")

access_dictionary_element()
```

```
Enter the key: banana
Value for key banana is: 2
```

```
def access_dictionary_element():
    my_dict = {"apple": 1, "banana": 2, "orange": 3}
    try:
        key = input("Enter the key: ")
        value = my_dict[key]
        print("Value for key", key, "is:", value)
    except KeyError:
        print("KeyError: Key not found in the dictionary.")

access_dictionary_element()
```

```
Enter the key: mango
KeyError: Key not found in the dictionary.
```