Ex.No.4	**USER DEFINED FUNCTIONS**	**Reg.No: URK23CS1261**
07.2.24		

4 A) Write a python program to check whether a number is a palindrome or not using a function

Aim: The objective of this program is to check whether a number is a palindrome or not using a function

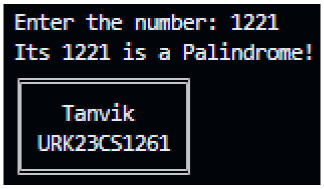
Algorithm:

```
Step 1: Start the program.
Step 2: Define a function named 'check_palindrome' that takes an 'item' as input and returns the reverse of the 'item'.
Step 3: Accept user input for a number and store it in the variable 'check_num' after stripping any leading or trailing spaces.
Step 4: Call the 'check_palindrome' function with 'check_num' as the argument and store the result in 'result'.
Step 5: Check if 'result' is equal to 'check_num'.
        -If true, print a message indicating that 'check_num' is a palindrome.
        -If false, print a message indicating that 'check_num' is not a palindrome.
Step 6: End the program.
```

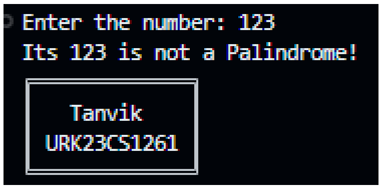
Program:

```
def check_palindrome(item):
    return item[::-1]
check_num = (input("Enter the number: ")).strip()
result = check_palindrome(check_num)
if result == check_num:
    print(f"Its {check_num} is a Palindrome!")
else:
    print(f"Its {check_num} is not a Palindrome!")
print(" ╔═══════════════╗\n║   Tanvik   ║\n║ URK23CS1261 ║\n╚═══════════════╝ ")
```

Output:



```
Enter the number: 1221
Its 1221 is a Palindrome!
╔═════════════╗
║   Tanvik    ║
║ URK23CS1261 ║
╚═════════════╝
```



```
Enter the number: 123
Its 123 is not a Palindrome!
╔═════════════╗
║   Tanvik    ║
║ URK23CS1261 ║
╚═════════════╝
```

Result: Thus, The program has successfully produced the desired output.

4 B) Write a python program to check Armstrong number using functions

Aim: The objective of this program is to check Armstrong number using functions

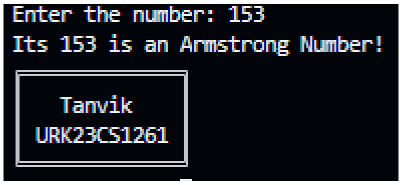
Algorithm:

```
Step 1: Start the program.
Step 2: Define a function named 'armstrong' that takes a string 'item' as input.
Step 3: Initialize 'result' to 0.
Step 4: Calculate the length of 'item' and store it in 'leng'.
Step 5: Iterate through each character 'i' in 'item'.
        a. Convert 'i' to an integer and raise it to the power of 'leng'.
        b. Add the result to 'result'.
Step 6: Return 'result'.
Step 7: Accept user input for a number and store it in the variable 'check_num' after stripping any leading or trailing spaces.
Step 8: Call the 'armstrong' function with 'check_num' as the argument and store the result in 'result'.
Step 9: Check if 'result' is equal to the integer value of 'check_num'.
        - If true, print a message indicating that 'check_num' is an Armstrong Number.
        - If false, print a message indicating that 'check_num' is not an Armstrong Number.
Step 10: End the program.
```

Program:

```
def armstrong(item):
    result = 0
    leng = len(item)
    for i in item:
        result += int(i) ** leng
    return result
check_num = (input("Enter the number: ")).strip()
result = armstrong(check_num)
if result == int(check_num):
   print(f"Its {check_num} is an Armstrong Number!")
else:
   print(f"Its {check_num} is not an Armstrong Number!")
print("╔═════════════╗\n║   Tanvik    ║\n║ URK23CS1261 ║\n╚═════════════╝")
```

Output:



```
Enter the number: 153
Its 153 is an Armstrong Number!
╔═════════════╗
║   Tanvik    ║
║ URK23CS1261 ║
╚═════════════╝
```

Result: Thus, The program has successfully produced the desired output.

4 C) Write python functions to print the no. of uppercase, lowercase, and whitespaces.

Aim: The objective of this program is to print the no. of uppercase, lowercase, and whitespaces

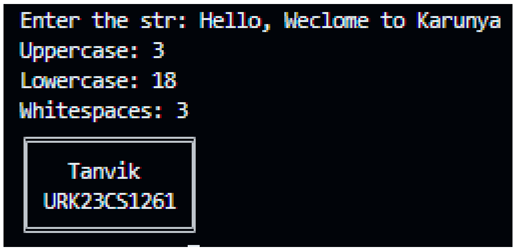
Algorithm:

```
Step 1: Start the program.
Step 2: Define a function 'check_space' taking a string 'sent' as input, initializing 'count' to 0.
Step 3: Iterate through each character 'char' in 'sent', incrementing 'count' if 'char' is a whitespace.
Step 4: Return 'count'.
Step 5: Define 'check_lower' similarly, counting lowercase letters.
Step 6: Define 'check_upper' similarly, counting uppercase letters.
Step 7: Accept user input for 'check_str'.
Step 8: Print counts of uppercase, lowercase, and whitespaces using respective functions.
Step 9: End the program.
```

Program:

```
def check_space(sent):
    count = 0
    for char in sent:
        if char.isspace():
            count += 1
    return count
def check_lower(sent):
    count = 0
    for char in sent:
        if char.islower():
            count += 1
    return count
def check_upper(sent):
    count = 0
    for char in sent:
        if char.isupper():
            count += 1
    return count
check_str = input("Enter the str: ")
print(f"Uppercase: {check_upper(check_str)}\nLowercase: {check_lower(check_str)}\nWhitespaces:
{check_space(check_str)}")
       print("╔═════════════╗\n║   Tanvik    ║\n║ URK23CS1261 ║\n╚═════════════╝")
```

Output:



```
Enter the str: Hello, Weclome to Karunya
Uppercase: 3
Lowercase: 18
Whitespaces: 3

╔═════════════╗
║   Tanvik    ║
║ URK23CS1261 ║
╚═════════════╝
```

Result: Thus, The program has successfully produced the desired output.

Ex.No.5 14.2.24	USAGE OF FILES OPERATIONS IN PYTHON	Reg.No: URK23CS1261

5 A) Write a Python Program to read a file’s entire content and store it back in another file in a reverse manner.

Aim: The objective of this program is to read a file’s entire content and store it back in another file in a reverse manner.

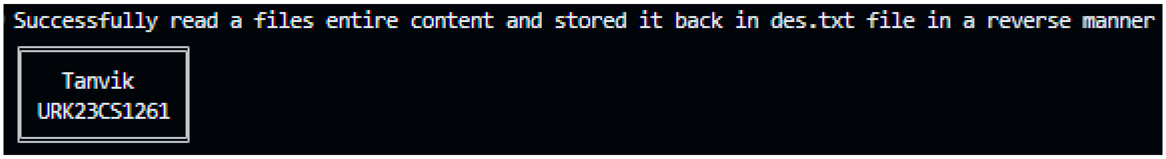
Algorithm:

Step 1: Start the program.
Step 2: Open the source file "src.txt" in read mode as 'srcfile'.
Step 3: Read the entire content of 'srcfile' and store it in 'reada'.
Step 4: Open the destination file "des.txt" in write mode as 'desfile'.
Step 5: Split the content of 'reada' into a list of words using the split() method with space as the delimiter, then reverse the list, and finally join the reversed list into a string separated by spaces.
Step 6: Write the reversed content to 'desfile'.
Step 7: Print the message "Successfully read a files entire content and stored it back in des.txt file in a reverse manner".
Step 8: Close both files.
Step 9: End the program.

Program:

```
with open("src.txt", "r") as srcfile:
    reada = srcfile.read()
    with open ("des.txt", "w") as desfile:
        desfile.write(" ".join((reada.split(" "))[::-1]))
        print("Successfully read a files entire content and stored it back in des.txt file in a reverse manner")
 print(" ╔══════════════╗ \n║   Tanvik    ║\n║ URK23CS1261 ║\n╚══════════════╝ ")
```

Output:



```
Successfully read a files entire content and stored it back in des.txt file in a reverse manner
╔═════════════╗
║   Tanvik    ║
║ URK23CS1261 ║
╚═════════════╝
```

Result: Thus, The program has successfully produced the desired output.

5 B) Write a python function to copy the content of one file into another file.

Aim: The objective of this program is to copy the content of one file into another file.

Algorithm:

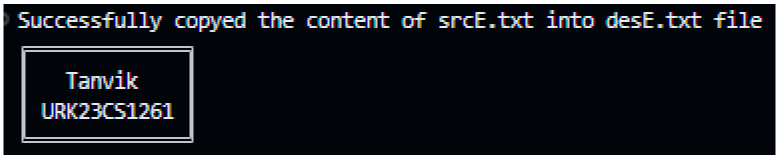
```
Step 1: Start the program.
Step 2: Define a function named 'copyed' that takes two parameters: 'src_file' and 'des_file'.
Step 3: Open the source file 'src_file' in read mode as 'srcfile'.
Step 4: Read the entire content of 'srcfile' and store it in 'reada'.
Step 5: Open the destination file 'des_file' in write mode as 'desfile'.
Step 6: Write the content of 'reada' to 'desfile'.
Step 7: Print a success message indicating that the content of 'src_file' has been copied to 'des_file'.
Step 8: Call the 'copyed' function with "srcE.txt" and "desE.txt" as arguments.
Step 9: Print a formatted block of text with a name and ID.
Step 10: End the program.
```

Program:

```
def copyed(src_file, des_file):
    with open(f"{src_file}", "r") as srcfile:
        reada = srcfile.read()
        with open (f"{des_file}", "w") as desfile:
            desfile.write(reada)
            print(f"Successfully copyed the content of {src_file} into {des_file} file")

copyed("srcE.txt", "desE.txt")
print(" ╔═════════════╗\n║   Tanvik    ║\n║ URK23CS1261 ║\n╚═════════════╝ ")
```

Output:



```
Successfully copyed the content of srcE.txt into desE.txt file
 ╔═════════════╗
 ║   Tanvik    ║
 ║ URK23CS1261 ║
 ╚═════════════╝
```

Result: Thus, The program has successfully produced the desired output.

5 C) Write a Python program to extract the word starting in vowels from a text file and print the same.

Aim: The objective of this program is to extract the word starting in vowels from a text file and print the same.

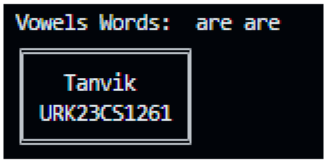
Algorithm:

```
Step 1: Start the program.
Step 2: Open the file "srcF.txt" in read mode as 'srcfile'.
Step 3: Read the entire content of 'srcfile', split it into a list of words using the split() method with space as the delimiter, and store it in 'reada'.
Step 4: Define a list 'vowels' containing the vowels 'a', 'e', 'i', 'o', and 'u'.
Step 5: Print "Vowels Words: " without a newline.
Step 6: Iterate through each word 'word' in 'reada'.
Step 7: Iterate through each vowel 'vow' in 'vowels'.
Step 8: Check if 'word' starts with 'vow'.
- If true, print 'word' without a newline and a space.
Step 9: Print a newline to move to the next line.
Step 10: Print a formatted block of text with a name and ID.
Step 11: End the program.
```

Program:

```
with open("srcF.txt", "r") as srcfile:
    reada = (srcfile.read()).split(" ")
    vowels = ["a", "e", "i", "o", "u"]
    print("Vowels Words: ", end = " ")
    for word in reada:
        for vow in vowels:
            if word.startswith(vow):
                print(word, end = " ")
print("\n╔═════════════╗\n║   Tanvik    ║\n║ URK23CS1261 ║\n╚═════════════╝")
```

Output:



```
Vowels Words:  are are

 ╔═════════════╗
 ║   Tanvik    ║
 ║ URK23CS1261 ║
 ╚═════════════╝
```

Result: Thus, The program has successfully produced the desired output.