|  |  |  |
| --- | --- | --- |
| **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:**            **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:**    **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:**    **Result:** Thus, The program has successfully produced the desired output. | | | |
| **Ex.No.5** | **USAGE OF FILES OPERATIONS IN PYTHON** | **Reg.No: URK23CS1261** |
| **14.2.24** |
| **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:   **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:**    **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:**    **Result:** Thus, The program has successfully produced the desired output. | | | | |