```
def count_word_frequency(text, word):
    text = text.lower()
    words = text.split()
    word_freq = 0

    for w in words:
        if w == word.lower():
            word_freq += 1

    return word_freq


def print_data_types(text):
    elements = text.split()
    for element in elements:
        print(f"Element: '{element}', Data Type:
{type(element).__name__}")


def count_characters(about):
    alphabet_count = 0
    numeric_count = 0
    special_count = 0

    for char in about:
        if char.isalpha():
            alphabet_count += 1

        elif char.isnumeric():
            numeric_count += 1
        else:
            special_count += 1

    return alphabet_count, numeric_count, special_count


while True:
    print("**************************__LAB EXERCISE
1__**************************")
    print("\n 1 ->  Domain ")
```

```
    print("\n 2 ->  Sets & Tuples")
    print("\n 3 -> Exit.")

print("**********************************************************************
*******")
    choice = int(input("\n\nEnter your choice : "))
    if (choice == 1):
        name = "Ansh Bhandari"
        dom = "University Student Management"
        classs = "1MCA-B"
        reg_no = 2347212
        course = "MCA"
        avg_marks = 90
        year = 2023
        about = "Hi, I'm Ansh Bhandari a student of 1 MCA-B
BATCH(2023-2025). I choosed my domain as University Student Management. \n
A good university management system ensures improved academic delivery,
working efficiency and better student achievements."
        list = ["Ansh", "Bhandari", 2347212, True,
                "University Student Management", "male", "marks", 520]

print("------------------------------------------------------------------------
-------------------------------------------------------------")
        print("Name :", name)
        print("Register No :", dom)
        print("Class :", classs)
        print("Course :", course)
        print("About:", about)

print("------------------------------------------------------------------------
-------------------------------------------------------------")

        print("\n\n Frequency of specific word")
        tar_word = str(input("\n Enter the target word :"))
        frequency = count_word_frequency(about, tar_word)

        print(f"\n The target word '{tar_word}' appears {frequency} times
in the text.\n")
        print("\n Datatypes of specific word \n")
        print_data_types(tar_word)
```

```
        print(f"\n The variable name '{name}' is of type: ", type(name))
        print(f"\n The variable domain '{dom}' is of type: ", type(dom))
        print(f"\n The variable class '{classs}' is of type: ",
type(classs))
        print(f"\n The variable Register No '{reg_no}' is of type: ",
type(reg_no))
        print(f"\n The variable Course '{course}' is of type: ",
type(course))
        print(f"\n The variable Average Marks '{avg_marks}' is of type: ",
type(avg_marks))
        print(f"\n The variable year '{year}' is of type: ", type(year))
        print(f"\n The variable name '{name}' is of type: ", type(name))
        print("\n The variable about is of type: ", type(about))
        print("\n The variable list is of type: ", type(list))

        print("\n\n Number of alphabets, numeric and other special symbols
")
        alphabet_count, numeric_count, special_count =
count_characters(about)
        print("\n Number of alphabets:", alphabet_count)
        print("\n Number of numeric characters:", numeric_count)
        print("\n Number of special symbols:", special_count)

        """
        University Management System Data Types Set

        Insights:
        1. Data Types in the Set:
            - Integers: Representing student IDs.
            - Floats: Used for representing GPA with decimal points.
            - Strings: Storing student names, course names & department
names.
   -        - Boolean: Representing enrollment status (True/False) for
course availability.

        2. Function Demonstration:
         - pop(): The `pop()` function removes and returns an arbitrary
element from the set. After using `pop()`, the set is updated, and the
removed element is displayed.
```

```
        - clear(): The `clear()` function removes all elements from the
set, hence making it an empty set. It is useful when you want to reset or
reuse the set with new data.
        - discard(): The `discard()` function is used to remove a
specific element from the set if it exists. It allows you to eliminate a
particular element without raising an error even if the element is not
present in the set.
        - del: The `del` keyword is used to delete the entire set. Once
the set is deleted, accessing it will raise a `NameError`.

        """

    elif (choice == 2):
        def university_management_system():
            data_types_set = {1, 8.5, "Ansh Bhandari", True,
                              "Pass", "A+", "Computer Science", "MCA"}

            popped_element = data_types_set.pop()
            print(f"Popped Element: {popped_element}")
            print(f"Updated Set after pop(): {data_types_set}")

            data_types_set.clear()
            print(f"Set after clear(): {data_types_set}")

            data_types_set = {1, 10.0, "Ansh Bhandari",
                              True, "Pass", "O", "Computer Science",
"MCA","COD","ADT","Python","C","Web Stack"}

            data_types_set.discard(10.0)
            print(f"Set after discarding (10.0)): {data_types_set}")

            #  del data_types_set
            #  try:
            #      print(data_types_set)
            #  except NameError as e:
            #      print("Set has been deleted.")
            #      print(f"Error Message: {e}")
            data = {1, 10.0, "Ansh Bhandari", True, "Pass", "O",
"Computer Science", "MCA","COD","ADT","Python","C","Web Stack"}
            print("----Descending Order----")
```

```
            sorted_set = sorted([str(x) for x in data_types_set],
reverse=True)
            print(f"Sorted Set (Descending order): {sorted_set}")

            print(f"Sorted Set (Descending order): {sorted_set}")


        def count_characters(domain):

            char_count = len(domain)
            return char_count


        def slicing_and_negative_indexing(domain):

            print("\n Original Domain:", domain)
            print("Slicing Operations:")
            print("1. First 5 characters:", domain[:5])
            print("2. Characters from index 3 to 9:", domain[3:10])
            print("3. Last 4 characters:", domain[-4:])
            print("4. Every second character:", domain[::2])
            print("5. Reverse the domain:", domain[::-1])

            print("\n Negative Indexing:\n")
            print("Last character:", domain[-1])
            print("Second last character:", domain[-2])
            print("Third last character:", domain[-3])
            print("Characters from index -6 to -2:", domain[-6:-1])


        if __name__ == "__main__":
            university_management_system()
            domain = "University Management System"
            character_count = count_characters(domain)
            print(f"The number of characters in the domain \"{domain}\"
is: {character_count}")
            slicing_and_negative_indexing(domain)


    elif choice == 3:
```

```
        break
    else:
        print("\nInvalid Input : Please try again")
```

Screenshots

```
PS E:\Ansh\MCA\1 TRI\Python> & C:/Users/anshb/AppData/Local/Programs/Python/Python310/python.exe "e:/Ansh/MCA/1 TRI/Python/py1.py"
**************************__LAB EXERCISE 1__**************************

 1 ->  Domain

 2 ->  Sets & Tuples

 3 -> Exit.
**************************************************************************


Enter your choice : 1
--------------------------------------------------------------------------------------------------------------------------------
Name : Ansh Bhandari
Register No : University Student Management
Class : 1MCA-B
Course : MCA
About: Hi, I'm Ansh Bhandari a student of 1 MCA-B BATCH(2023-2025). I choosed my domain as University Student Management.
 A good university management system ensures improved academic delivery, working efficiency and better student achievements.
--------------------------------------------------------------------------------------------------------------------------------


 Frequency of specific word

 Enter the target word :ansh

 The target word 'ansh' appears 1 times in the text.
```

```
 Datatypes of specific word

Element: 'ansh', Data Type: str

 The variable name 'Ansh Bhandari' is of type:  <class 'str'>

 The variable domain 'University Student Management' is of type:  <class 'str'>

 The variable class '1MCA-B' is of type:  <class 'str'>

 The variable Register No '2347212' is of type:  <class 'int'>

 The variable Course 'MCA' is of type:  <class 'str'>

 The variable Average Marks '90' is of type:  <class 'int'>

 The variable year '2023' is of type:  <class 'int'>

 The variable name 'Ansh Bhandari' is of type:  <class 'str'>

 The variable about is of type:  <class 'str'>

 The variable list is of type:  <class 'list'>


 Number of alphabets, numeric and other special symbols

 Number of alphabets: 187

 Number of numeric characters: 9

 Number of special symbols: 44
```

```
Enter your choice : 2
Popped Element: 1
Updated Set after pop(): {'Pass', 'Computer Science', 8.5, 'Ansh Bhandari', 'MCA', 'A+'}
Set after clear(): set()
Set after discarding (10.0)): {1, 'Pass', 'C', 'ADT', 'O', 'Web Stack', 'Python', 'Ansh Bhandari', 'COD', 'MCA', 'Computer Science'}
----Descending Order----
Sorted Set (Descending order): ['Web Stack', 'Python', 'Pass', 'O', 'MCA', 'Computer Science', 'COD', 'C', 'Ansh Bhandari', 'ADT', '1']
Sorted Set (Descending order): ['Web Stack', 'Python', 'Pass', 'O', 'MCA', 'Computer Science', 'COD', 'C', 'Ansh Bhandari', 'ADT', '1']
The number of characters in the domain "University Management System" is: 28

 Original Domain: University Management System
Slicing Operations:
1. First 5 characters: Unive
2. Characters from index 3 to 9: versity
3. Last 4 characters: stem
4. Every second character: Uiest aaeetSse
5. Reverse the domain: metsyS tnemeganaM ytisrevinU

 Negative Indexing:

Last character: m
Second last character: e
Third last character: t
Characters from index -6 to -2: Syste
```