**Modern Education Society’s**

**College of Engineering, Pune**

|  |
| --- |
| **NAME OF STUDENT: Prathamesh Kalyan Sable CLASS: SE Comp 1** |
| **SEMESTER/YEAR: Sem – 3 / 2022 ROLL NO: F21111015** |
| **DATE OF PERFORMANCE: 07/09/2022 DATE OF SUBMISSION: 14/09/2022** |
| **EXAMINED BY: Mrs. N.R. Mhaske EXPERIMENT NO: DSL A-02** |

###### TITLE: Perform various operations on array

**PROBLEM STATEMENT:**  Write a Python program to store marks scored in subject “Fundamental of Data Structure” by N students in the class. Write functions to compute following:

1. The average score of class
2. Highest score and lowest score of class
3. Count of students who were absent for the test
4. Display mark with highest frequency

### **OBJECTIVES:**

1. To understand structure of Array.
2. To understand how create, display and perform various operations on array.

### **OUTCOMES:**

1. To analyze the problems to apply suitable algorithm and data structure.
2. To discriminate the usage of various data structures in approaching the problem solution.
3. To understand concept of linear data structure

**PRE-REQUISITES:**

* 1. Knowledge of python programming
  2. Knowledge of array data structure

**APPARATUS:**

Computer Machine, python3 installed, etc.

**QUESTIONS:**

1. What is static and dynamic memory allocation?
2. Explain difference between list and array in python with an example.

**SOURCE CODE:**

# global lists for present absent total students

marks\_of\_stu = []

present\_stu\_marks = []

absent\_stu\_list = []

def add\_marks():

    # addign marks to list

    n = int(input("Enter total number of students :"))

    print("Enter A/a/-1 for Absent student")

    for i in range(n):

        mark = input(f"Enter marks of student with roll number {i+1} :")

        marks\_of\_stu.append(mark)

    # separating absent and present sudents

    for mark in marks\_of\_stu:

        if mark in ('a','A','-1'):

            absent\_stu\_list.append(mark)

        else:

            present\_stu\_marks.append(int(mark))

def find\_average():

    sum = 0

    n = 0

    for mark in present\_stu\_marks:

        sum += mark

        n+=1

    print("Average of the class is %.2f"%(sum/n))

def display\_min\_max():

    # initialize lowest and highest marks to index 0

    lowest = highest = present\_stu\_marks[0]

    for mark in present\_stu\_marks:

        if mark<lowest:

            lowest = mark

        if mark>highest:

            highest=mark

    print("Highest score of the class is", highest)

    print("Lowest score of the class is", lowest)

def find\_max\_frequency():

    marks = []

    freq = []

    for i in present\_stu\_marks:

        if i not in marks:

            marks.append(i)

            freq.append(1)

        else:

            index = marks.index(i)

            freq[index] += 1

    max\_mark = []

    max = freq[0]

    for i in range(len(freq)):

        if freq[i]>max:

            max = freq[i]

    for i in range(len(freq)):

        if freq[i] == max:

            max\_mark.append(marks[i])

    if len(max\_mark) == 1:

        print("The marks with highest frequency is", max\_mark[0])

    else:

        print("The marks with highest frequency are", max\_mark)

add\_marks()

while True:

    print("--MENU--")

    print("1. The Average score ")

    print("2. Highest and Lowest score")

    print("3. Count of students who were absent for the test")

    print("4. Display mark with highest frequency")

    print("5. Re Enter the marks")

    print("6. Exit the code")

    choice = input("Enter your choice :")

    if choice == '1':

        find\_average()

    elif choice == '2':

        display\_min\_max()

    elif choice == '3':

        print("The number of students absent for the test are", len(absent\_stu\_list))

    elif choice == '4':

        find\_max\_frequency()

    elif choice == '5':

        add\_marks()

    elif choice == '6':

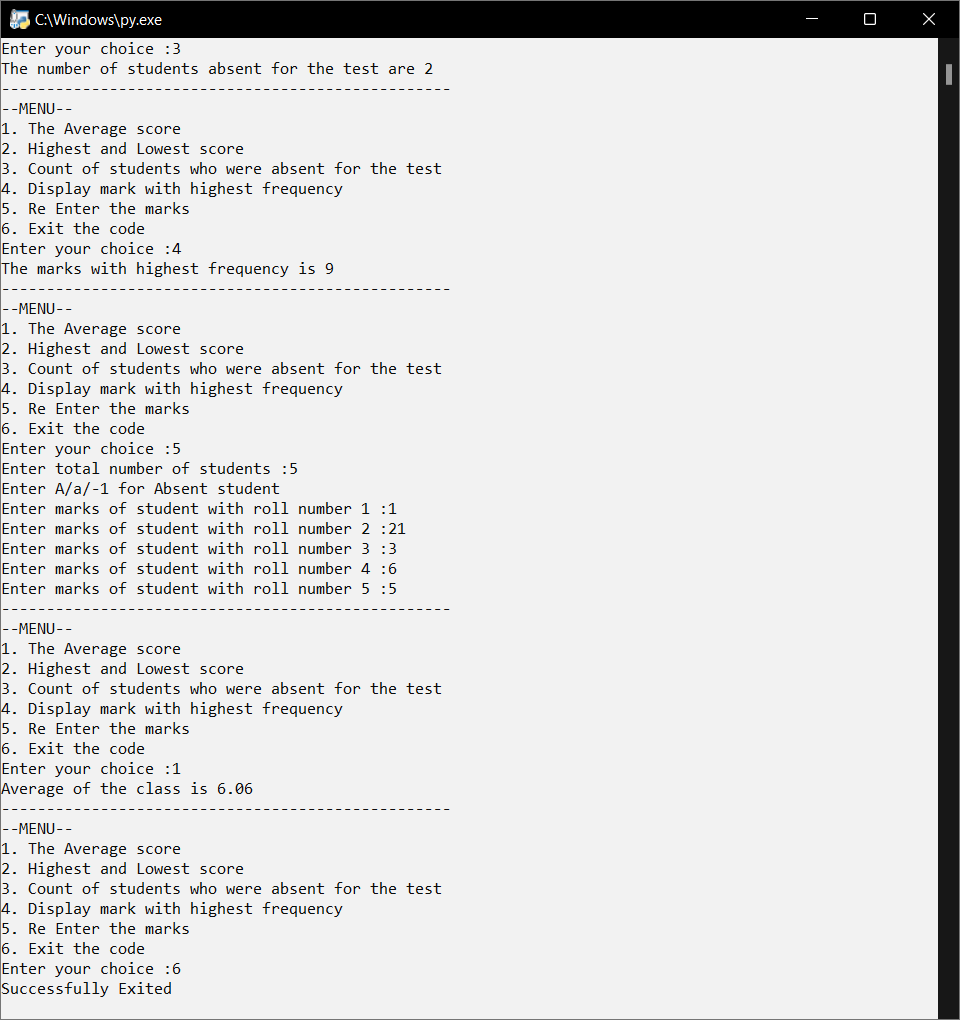
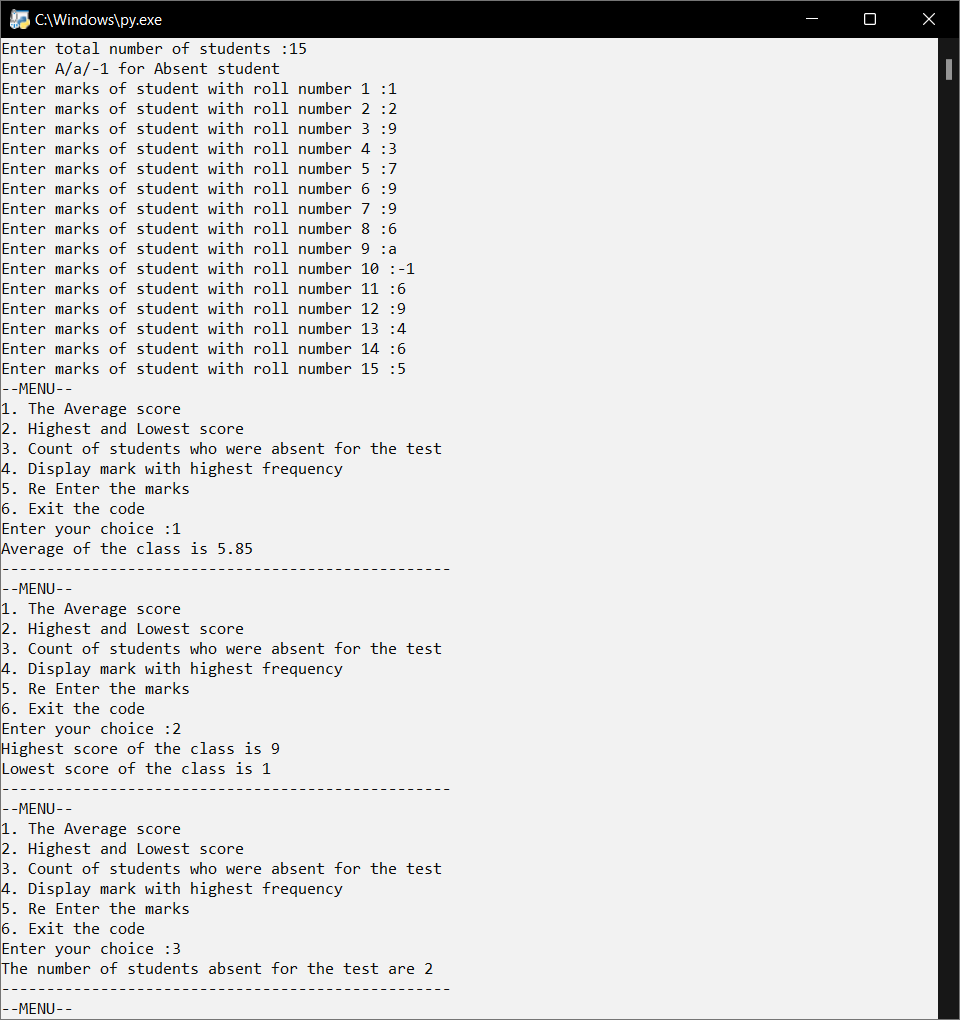
        break

    else:

        print("Please Enter a valid choice.")

    print("-"\*50)

print("Successfully Exited")

**OUTPUT:**