NAME: RABIA BATOOL

ROLL NO. : 2022-BSE-064

LAB#1

SUBJECT : DATA STRUCTURE AND ALGORITHIMS

SUBMITTED TO: SIR REHAN

Lab#1:

Write the prototype of a function named DS() which accepts an array of integers, a

pointer to double, a float by reference and returns a character.

Answer:

```
Char ds(int arr[],double *ptr,float &ref)
```

2 Write statements as directed:

```
int a[]={1,2,3,4,5};

int *p;

p = a;

//Print the elements of array using the mentioned notation:

for (int i = 0; i < 5; i++)

{

//Subscript notation with name of array

Cout<<"elements"<<a[i]<<endl;

//Subscript notation with pointer 'p'

Cout<<"elements"<<p[i]<<endl;

//Offset notation using array name

Cout<<elements using off set of array name"<<*(a+i)<<endl;

//Offset notation using pointerp

Cout<<elements using off set of pointer name"<<*(p+i)<<endl;
```

3. What is the difference between function overloading and function overriding?

Function overloading: refers to the ability to define **multiple functions** in the same class or scope with the same name **but different parameter lists** (i.e., a different number or type of parameters). It allows you to create functions with the same name that perform different tasks based on their parameter signatures.

Function overriding: is a feature of inheritance in object-oriented programming. It allows a subclass to provide **a specific implementation** of a function that is already defined in its superclass. Both the base class (superclass) and the derived class (subclass) have functions with the same name, same parameters, and the same return type.

4. Write C++ statement(s) to allocate space for 10 doubles (using dynamic memory allocation).

```
Double  *ptr;
```

Ptr=new double[10];

5 Study the given program and determine what the program is intended to do. (Hint: Dry

run the program with array {1,2,1,2,3} and analyze the output).

Consider we have entered 1,2,1,2,3 in array.first the program will compare the first index value with the second index value if both are save it will remover the dublicate element and print the array unique elements so, the main purpose of program is to remove **dublicate elements** of array.

Write a C++ function which accepts an array of integers and the size of the array and finds :

a. Sum of the elements in the array

b. Average of the array elements

c. Minimum and maximum values in the array

In the main program, declare an array of 7 integers using dynamic memory allocation and call

the aforementioned function. Display the output of the function within the main.

include <iostream>

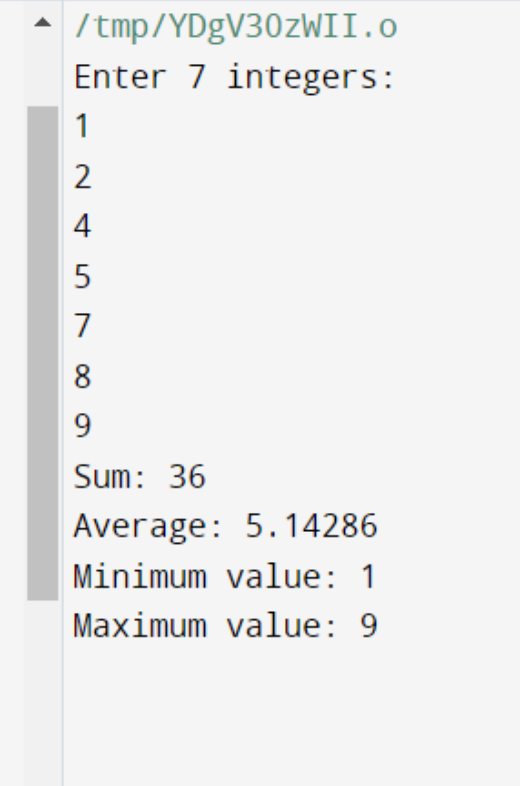
```
using namespace std;
void Array(int arr[], int size, int& sum, double& average, int& minVal, int& maxVal) {
   if (size <= 0) {
      cout << "Array is empty. Cannot calculate statistics." << endl;
      return;
   }
   sum = 0;
   minVal = arr[0];
   maxVal = arr[0];

   for (int i = 0; i < size; i++) {
      sum += arr[i];
      if (arr[i] < minVal) {
```

```
            minVal = arr[i];
        }
        if (arr[i] > maxVal) {
            maxVal = arr[i];
        }
    }
    average = (sum) / size;
}

int main() {
    int size = 7;
    int* arr = new int[size];
    std::cout << "Enter " << size << " integers:" << std::endl;
    for (int i = 0; i < size; i++) {
        cin >> arr[i];
    }
    int sum, minVal, maxVal;
    double average;
    findArrayStats(arr, size, sum, average, minVal, maxVal);
    cout << "Sum: " << sum << endl;
    cout << "Average: " << average << endl;
    :cout << "Minimum value: " << minVal << endl;
    :cout << "Maximum value: " << maxVal << endl;
    delete[] arr;

    return 0;
}
```



Exercise 2

Write a program with a function which accepts an array of integers and a key value. The function should return the sum of all the multiples of the key value in the array. For example, for the array {1, 4, 10, 12, 15, 20, 22} and the key value 5, the function should return the sum 10+15+20.

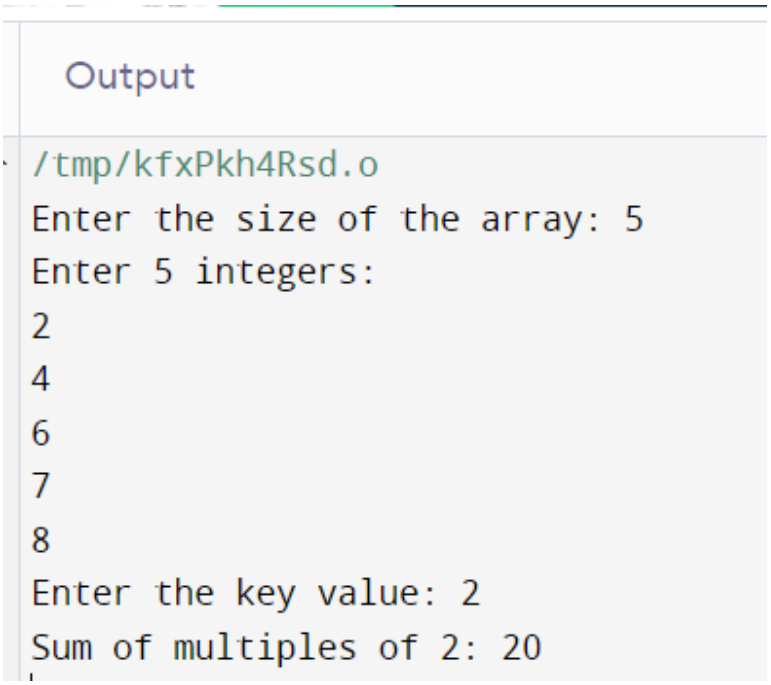
Task#2

```
#include <iostream>


int sumOfMultiples(int arr[], int size, int key) {
   int sum = 0;
   for (int i = 0; i < size; i++) {
      if (arr[i] % key == 0) {
         sum += arr[i];
      }
```

```
    }
    return sum;
int main() {
    int size;    std::cout << "Enter the size of the array: ";
    std::cin >> size;
    int* arr = new int[size]
    scout << "Enter " << size << " integers:" << endl;
    for (int i = 0; i < size; i++) {
scin >> arr[i];
    }
 int key;
    std::cout << "Enter the key value: ";
    std::cin >> key;
    int result = sumOfMultiples(arr, size, key);
    std::cout << "Sum of multiples of " << key << ": " << result << std::endl;
    delete[] arr;

    return 0;
}
```



Task #3

Exercise 3

Create a class Matrix to model 2x2 matrices. Provide a default , parameterized constructor,get_input method to assign values to the matrix. Using a member function,

Lab 1

overload the '+' operator to add two matrices. Likewise, overload the '~' operator to find the determinant of the matrix. Also provide a display() member function to print the matrix. In the main program, create an object of class Matrix and call its member functions.

```
// Online C++ compiler to run C++ program online
#include <iostream>
using namespace std;


class Matrix {
```

```
private:
    int mat[2][2];
public:
    Matrix() {
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                mat[i][j] = 0;
            }
        }
    }
    Matrix(int a, int b, int c, int d) {
        mat[0][0] = a;
        mat[0][1] = b;
        mat[1][0] = c;
        mat[1][1] = d;
    }
    void get_input() {
        std::cout << "Enter matrix values (row-wise):" << std::endl;
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                std::cin >> mat[i][j];
            }
        }
    }
  Matrix operator+(const Matrix& other) {
        Matrix result;
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                result.mat[i][j] = mat[i][j] + other.mat[i][j];
```

```
        }
      }
      return result;
   }
   int operator~() {
      return (mat[0][0] * mat[1][1] - mat[0][1] * mat[1][0]);
   }

   void display() {
cout << "Matrix:" << endl;
      for (int i = 0; i < 2; i++) {
         for (int j = 0; j < 2; j++) {
            cout << mat[i][j] << " ";
         }
cout << std::endl;
      };

int main() {
   Matrix matrix1, matrix2, result;
   std::cout << "Enter values for the first matrix:" << std::endl;
   matrix1.get_input();
  std::cout << "Enter values for the second matrix:" << endl;
   matrix2.get_input();
   result = matrix1 + matrix2;
   std::cout << "Result of addition:" << std::endl;
   result.display();
   int determinant = ~matrix1;
   std::cout << "Determinant of the first matrix: " << determinant << std::endl;
```

```
    return 0;
}
```

Output

```
/tmp/FIZ1sc8D6v.o
Enter values for the first matrix:
Enter matrix values (row-wise):
1
2
3
4
Enter values for the second matrix:
Enter matrix values (row-wise):
5
4
4
5
Result of addition:
Matrix:
6 6
7 9
Determinant of the first matrix: -2
```