ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

Prepared by:

Name	CMS ID	Class	Section
Muhammad Asim Shah	470574	ME-15	C

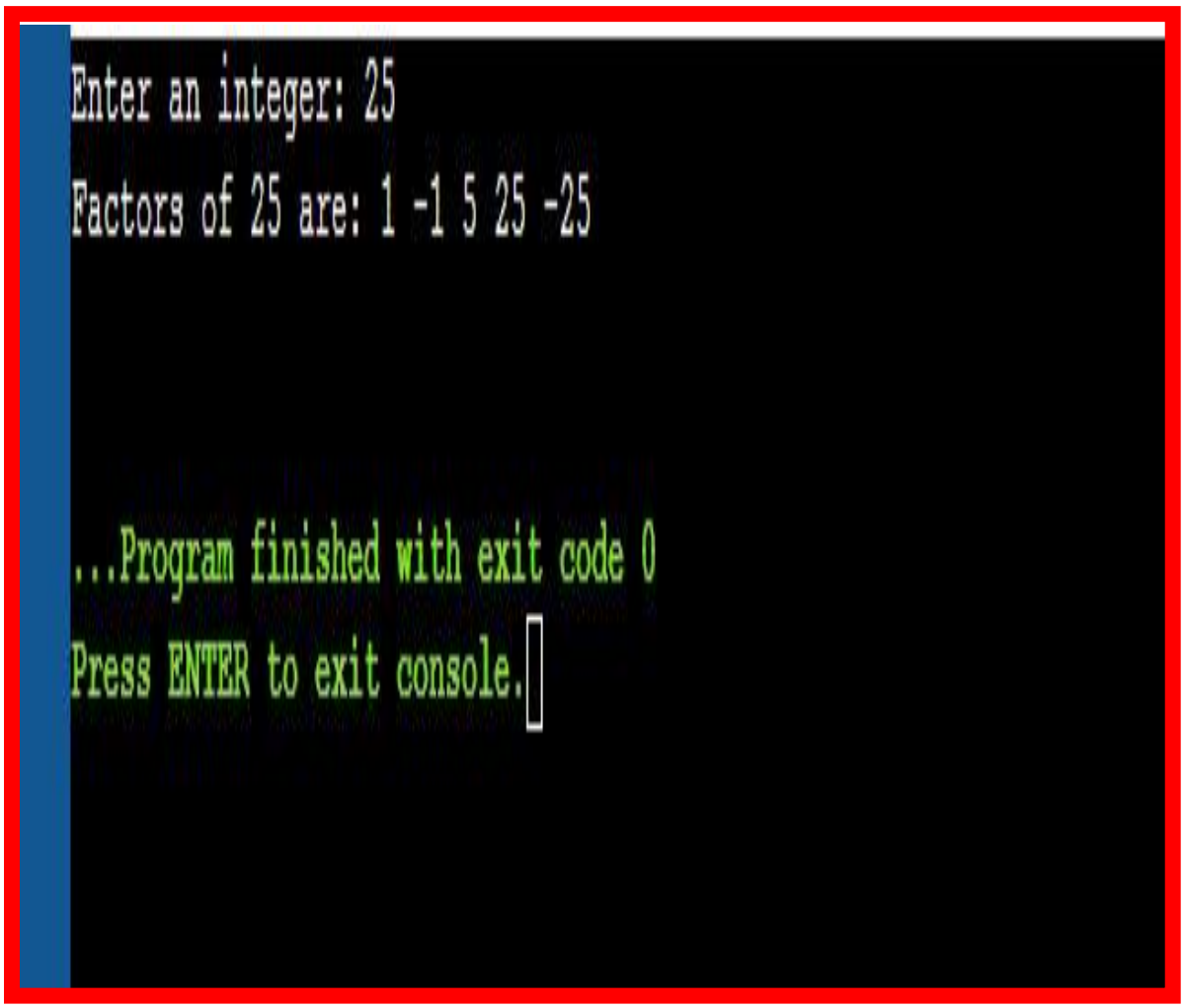
Task 1: Write a C++ program to display factors of a number using for loops.

Input:

```
#include <iostream>

using namespace std;


int main() {

   int num;


   cout << "Enter an integer: ";

   cin >> num;


   cout << "The Factors of " << num << " are: ";


   for (int i = 1; i <= num; ++i) {

      if (num % i == 0) {

         cout << i << " ";
```

```
    }
  }

  cout << endl;

  return 0;
}
```

Output:



ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

Task 2: Write output to the following code.

```
#include <iostream>

int main() {
    int x = 5;
    int y = 10;

    if (x == 5)
        if (y == 10)
            std::cout << "x is 5 and y is 10" << std::endl;
    else
        std::cout << "x is not 5" << std::endl;

    return 0;
}
```

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

Output:

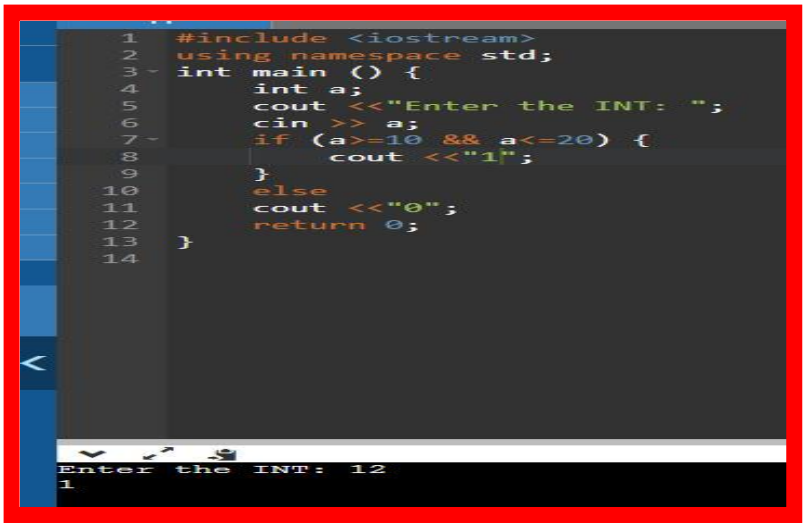
Task 3: Write a C++ program, take an integer value from user and check if it's greater than 10 and less than equal to 20. Print 1 if yes and print 0 if no. Use appropriate datatype for output.

Input:

```
#include <iostream>

using namespace std;

int main () {

   int a;

   cout <<"Enter the INT: ";

   cin >> a;

   if (a>=10 && a<=20) {

      cout <<"Yes";
   }
   else
```

```
    cout <<"No";

    return 0;
}
```

Output:



Task 4: Write a C++ program that uses a **while** loop to find the largest prime number less than a given positive integer **N**. Your program should take the value of **N** as input from the user and then find the largest prime number less than or equal to **N**. You are not allowed to use any library or pre-existing functions to check for prime numbers.

Input:

```
#include <iostream>

using namespace std;

int main() {

    int n;
```

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

```
    cout << "Enter an integer N: ";

    cin >> n;

    while (n > 1) {

        bool prime = true;

        for (int i = 2; i < n; ++i) {

            if (n % i == 0) {

                prime = false;

                break;
            }
        }

        if (prime) {

            cout << "Largest prime number less than or equal to N is : " << n << endl;

            break;
        }

        --n;
    }

    return 0;
}
```

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

Output:

```
int main() {
    int n;

    cout << "Enter an integer N: ";
    cin >> n;
    while (n > 1) {
        bool prime = true;

        for (int i = 2; i < n; ++i) {
            if (n % i == 0) {
                prime = false;
                break;
            }
        }

        if (prime) {
            cout << "Largest prime number less than or equal to N is : " << n << endl;
            break;
        }

        --n;
    }
```

```
input
Enter an integer N: 45
Largest prime number less than or equal to N is : 43


...Program finished with exit code 0
Press ENTER to exit console.
```


Task 5: Write a C++ program, take two string as input from user and check if both strings are equal or not. If they are equal make them unequal by rotating string. e.g., Hello is turned into olleH etc.

Input:

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

```
#include <iostream>

#include <cstring>

using namespace std;

void rotateCharArray(char arr[], int length) {

    char temp = arr[0];

    for (int i = 0; i < length - 1; ++i) {

        arr[i] = arr[i + 1];

    }
    arr[length - 1] = temp;
}

int main() {

    const int MAX_LENGTH = 100;


    char str1[MAX_LENGTH], str2[MAX_LENGTH];

    cout << "Enter the first string: ";
    cin >> str1;

    cout << "Enter the second string: ";

    cin >> str2;

    if (strcmp(str1, str2) == 0) {

        rotateCharArray(str1, strlen(str1));

        cout << "After rotation, the first string is: " << str1 << endl;
        cout << "The second string remains unchanged: " << str2 << endl;

    } else {
```

```
        cout << "The entered strings are already not equal." << endl;
    }

    return 0;
}
```

Output:

```
#include <iostream>
#include <cstring>
using namespace std;

void rotateCharArray(char arr[], int length) {
    char temp = arr[0];
    for (int i = 0; i < length - 1; ++i) {
        arr[i] = arr[i + 1];
    }
    arr[length - 1] = temp;
}

int main() {
    const int MAX_LENGTH = 100;

    char str1[MAX_LENGTH], str2[MAX_LENGTH];

    cout << "Enter the first string: ";
    cin >> str1;

    cout << "Enter the second string: ";
    cin >> str2;

    if (strcmp(str1, str2) == 0) {
```

input

```
Enter the first string: dolol
Enter the second string: dolol
After rotation, the first string is: olold
The second string remains unchanged: dolol
```

BANDICAM

Task 6: Perform division in C++ **without** / using **for** loops. You can use / only to display the final results. Your dividend must be greater than divisor.

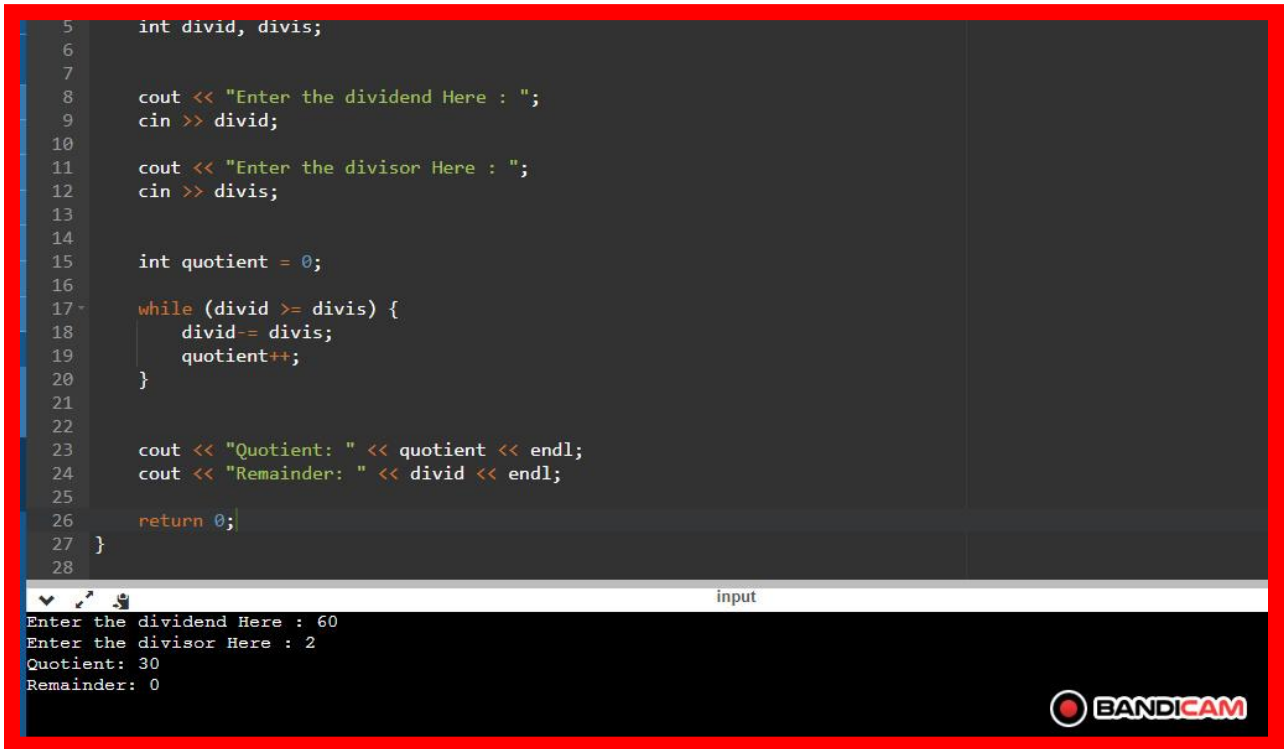
ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

Input:

```
#include <iostream>

using namespace std;

int main() {

    int divid, divis;


    cout << "Enter the dividend Here : ";

    cin >> divid;

    cout << "Enter the divisor Here : ";

    cin >> divis;


    int quotient = 0;

    while (divid >= divis) {

        divid-= divis;

        quotient++;
    }


    cout << "Quotient: " << quotient << endl;

    cout << "Remainder: " << divid << endl;

    return 0;


}
```

Output:

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB



Task 7: Write a C++program for a string which may contain lowercase and uppercase characters. The task is to remove all duplicate characters from the string and find the resultant string.

Input:

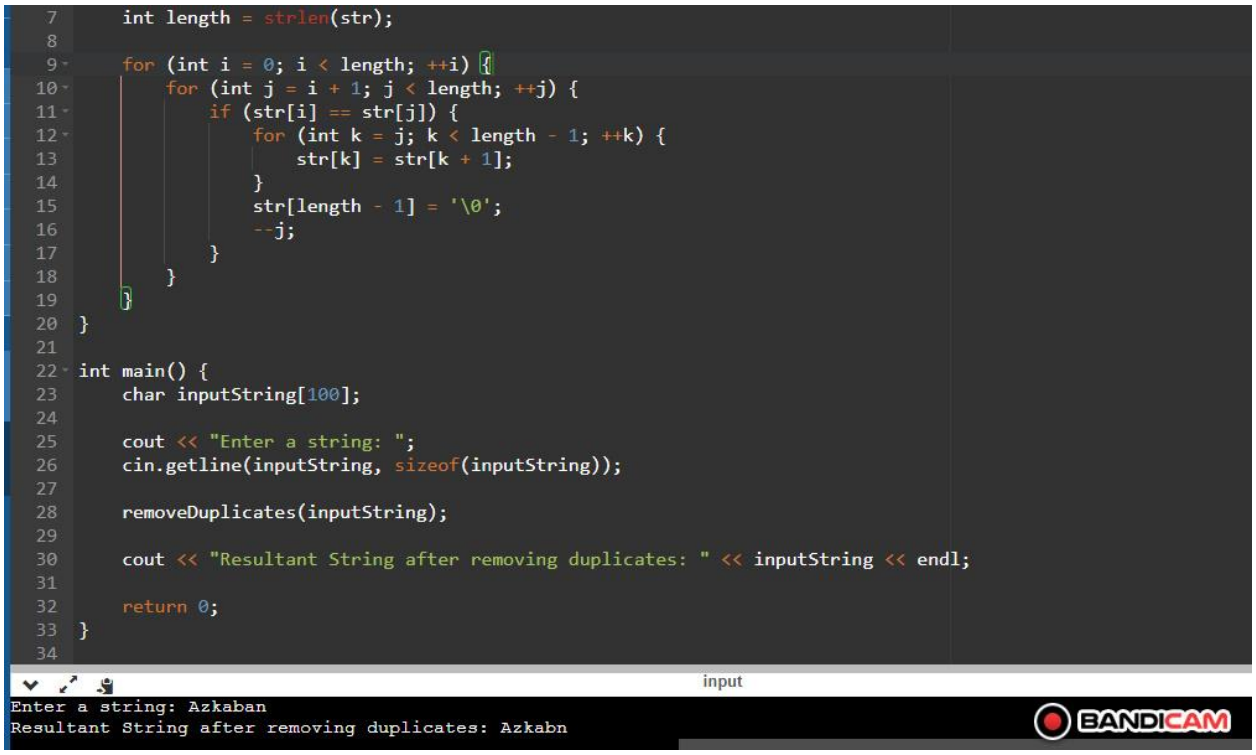
```
#include <iostream>

#include <cstring>


using namespace std;


void removeDuplicates(char str[]) {

    int length = strlen(str);
```

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

```
    for (int i = 0; i < length; ++i) {
        for (int j = i + 1; j < length; ++j) {
            if (str[i] == str[j]) {
                for (int k = j; k < length - 1; ++k) {
                    str[k] = str[k + 1];
                }
                str[length - 1] = '\0';
                --j;
            }
        }
    }
}

int main() {
    char inputString[100];

    cout << "Enter a string: ";
    cin.getline(inputString, sizeof(inputString));

    removeDuplicates(inputString);

    cout << "Resultant String after removing duplicates: " << inputString << endl;
```

```
    return 0;
}
```

Output:



```
    int length = strlen(str);

    for (int i = 0; i < length; ++i) {
        for (int j = i + 1; j < length; ++j) {
            if (str[i] == str[j]) {
                for (int k = j; k < length - 1; ++k) {
                    str[k] = str[k + 1];
                }
                str[length - 1] = '\0';
                --j;
            }
        }
    }
}

int main() {
    char inputString[100];

    cout << "Enter a string: ";
    cin.getline(inputString, sizeof(inputString));

    removeDuplicates(inputString);

    cout << "Resultant String after removing duplicates: " << inputString << endl;

    return 0;
}
```

```
input
Enter a string: Azkaban
Resultant String after removing duplicates: Azkabn
```


Task 8: Suppose an integer array a[5] = {1,2,3,4,5}. Add more elements to it and display them in C++.

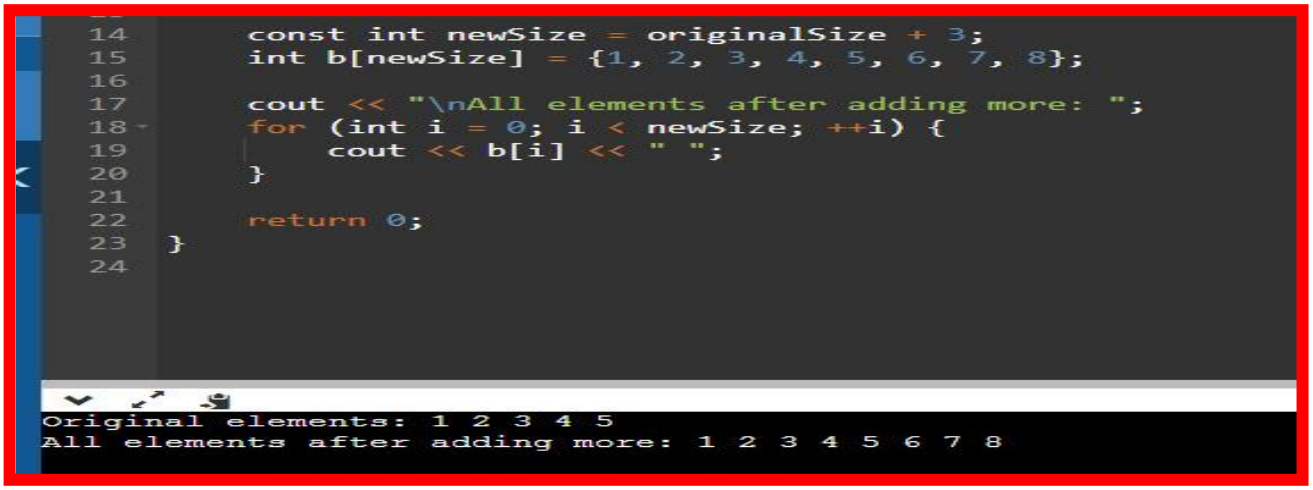
Input:

```
#include <iostream>

using namespace std;

int main() {

    const int originalSize = 5;

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

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

```
    cout << "Original elements: ";
    for (int i = 0; i < originalSize; ++i) {
        cout << a[i] << " ";
    }

    const int newSize = originalSize + 3;
    int b[newSize] = {1, 2, 3, 4, 5, 6, 7, 8};

    cout << "\nAll elements after adding more: ";
    for (int i = 0; i < newSize; ++i) {
        cout << b[i] << " ";
    }
  return 0;
}
```

Output:



ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

Task 9: Given an integer array and an integer **X**. Find if there's a triplet in the array which sums up to the given integer **X.**

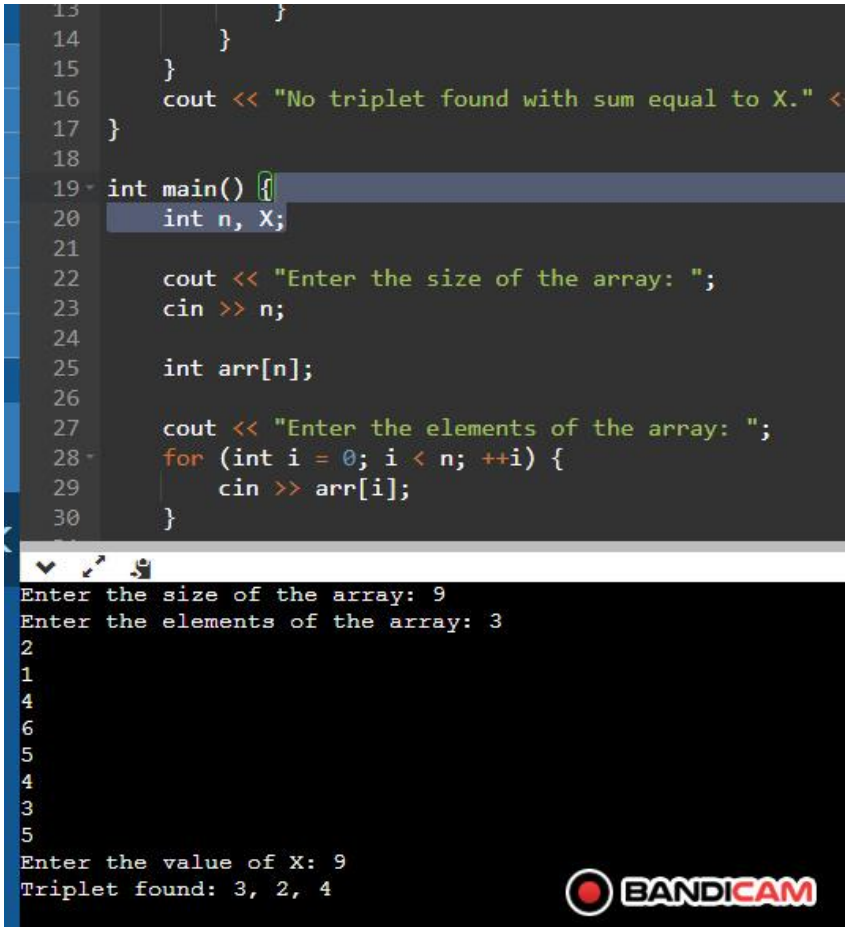
Input:

```
#include <iostream>


using namespace std;


void findTriplet(int arr[], int n, int X) {
    for (int i = 0; i < n - 2; ++i) {
        for (int j = i + 1; j < n - 1; ++j) {
            for (int k = j + 1; k < n; ++k) {
                if (arr[i] + arr[j] + arr[k] == X) {
                    cout << "Triplet found: " << arr[i] << ", " << arr[j] << ", " << arr[k] << endl;
                    return;
                }
            }
        }
    }
    cout << "No triplet found with sum equal to X." << endl;
}


int main() {
```

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

```
    int n, X;

    cout << "Enter the size of the array: ";
    cin >> n;

    int arr[n];

    cout << "Enter the elements of the array: ";
    for (int i = 0; i < n; ++i) {
        cin >> arr[i];
    }

    cout << "Enter the value of X: ";
    cin >> X;

    findTriplet(arr, n, X);

    return 0;
}
```

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB



Task 10: Implement Bubble Sort on an array of 6 integers.

Input:

```
#include <iostream>

#include <algorithm>


using namespace std;
```

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

```
int main() {
    const int size = 6;
    int array[size];

    cout << "Enter " << size << " integers for the array:" << endl;
    for (int i = 0; i < size; ++i) {
        cin >> array[i];
    }

    sort(array, array + size);

    cout << "Sorted array using std::sort: ";
    for (int i = 0; i < size; ++i) {
        cout << array[i] << " ";
    }

    return 0;
}
```

ASSIGNMENT 1
FUNDAMENTALS OF PROGRAMMING
CS-114 LAB

Output:

```cpp
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    const int size = 6;
    int array[size];

    cout << "Enter " << size << " integers for the array:" << endl;
    for (int i = 0; i < size; ++i) {
        cin >> array[i];
    }


    sort(array, array + size);

    cout << "Sorted array using std::sort: ";
    for (int i = 0; i < size; ++i) {
        cout << array[i] << " ";
    }

    return 0;
}
```

```
input
Enter 6 integers for the array:
7
0
4
3
2
1
Sorted array using std::sort: 0 1 2 3 4 7
```

Assignment 1
Fundamentals of Programming
CS-114 Lab