ASSIGNMENT 1

TAHA MUZAMMIL

456279

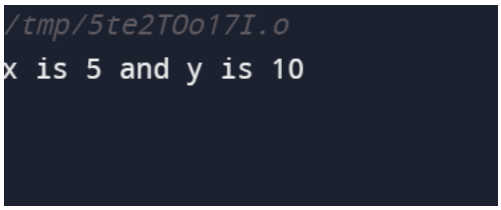
TASK 1

```
#include <iostream>
using namespace std;
int main()
{
   // Declaring the variables
   int n;
   cout<< "enter any positive whole number: ";
   cin>>n;
   // show factors with for loop
   cout << "Factors of " << n << " : ";
   for (int i = 1; i <= n; ++i) {
      // Checking for i as a factor
      if (n % i == 0) {
         cout << i << " ";
      }
   }
}
```

```
Output
/tmp/5te2TOo17I.o
enter any positive whole number: 10
Factors of 10 : 1 2 5 10
```

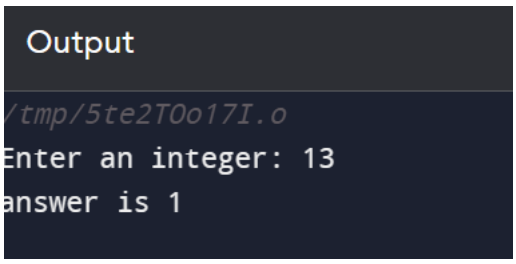
Task 2

```
#include <iostream>
using namespace std;
int main() {
 int x = 5;
 int y = 10;
 if (x == 5)
 if (y == 10)
 cout<< "x is 5 and y is 10" << std::endl;
 else
 cout<< "x is not 5" << std::endl;
 return 0;
}
```



Task 3

```
#include <iostream>
using namespace std;
 int main()
 {
   // Declare variables
   int num;
   // Prompt user for input
   cout << "Enter an integer: ";
   cin >> num;
   // putting in condition to check
   int x = (num > 10 && num <= 20) ? 1 : 0;
   cout << "answer is " << x << endl;
}
```



Task 4

```
#include <iostream>
using namespace std;
// checking if number is prime
bool isPrime(int num) {
    if (num <= 1) {
        return false;
    }

    for (int i = 2; i * i <= num; ++i) {
        if (num % i == 0) {
            return false;
        }
    }

    return true;
}

int main() {
    int N;
    cout << "Enter any positive integer N: ";
    cin >> N;

    // Find the largest prime number less than or equal to N
    while (N > 1) {
        if (isPrime(N)) {
            cout << "Largest prime number less than or equal to N: " << N << endl;
            break;
        }
```

```
        --N;
    }

    if (N <= 1) {
        cout << "No prime number within N" << endl;
    }
}
```

Output

```
/tmp/5te2TOo17I.o
Enter any positive integer N: 11
Largest prime number less than or equal to N: 11
```

Task 5

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

// Function to rotate a string
std::string rotateString(const std::string& str) {
    return str.substr(1) + str[0];
}

int main()
{
    // Input two strings from the user
    string a, b;

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

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

    // Check if the strings are equal
    if (a == b) {
        cout << "The strings are equal.\n";

        // Rotate one of the strings to make them unequal
        b = rotateString(b);

        cout << "After rotating the second string: " << b << "\n";
```

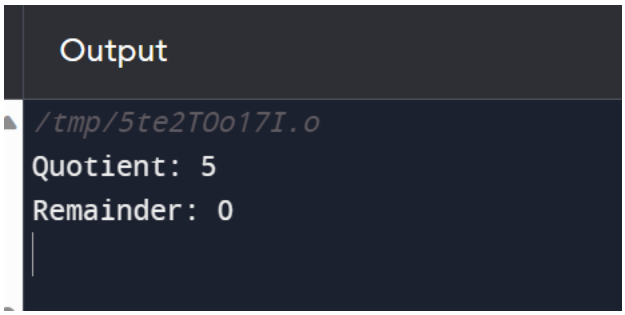
```
    } else {
        cout << "The strings are not equal.\n";
    }
}
```

Output

```
/tmp/5te2TOo17I.o
Enter the first string: taha
Enter the second string: taha
The strings are equal.
After rotating the second string: ahat
```

Task 6

```
#include <iostream>
using namespace std;
int main()
{
   int dividend = 30;
   int divisor = 6;

   // Ensure dividend is greater than divisor
   if (dividend < divisor) {
      cout << "Dividend must be greater than divisor." << endl;
      return 1;
   }

   int quotient = 0;
   while (dividend >= divisor) {
      dividend -= divisor;
      quotient++;
   }

   // show final answer
   cout << "Quotient: " << quotient << endl;
   cout << "Remainder: " << dividend << endl;
}
```



Task 7

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

   string removeDuplicates(const std::string& input) {
   string result;
   bool charSet[256] = {false};

   for (char ch : input) {
      if (!charSet[static_cast<unsigned char>(ch)]) {
         result += ch;
         charSet[static_cast<unsigned char>(ch)] = true;
      }
   }

   return result;
}

int main() {
   string input;

   // Input string
   cout << "Enter a string: ";
   getline(cin, input);

   // Remove duplicates
   string result = removeDuplicates(input);
```

```
    // Display result
    cout << "Resultant string after removing duplicates: " << result << endl;

    return 0;
}
```

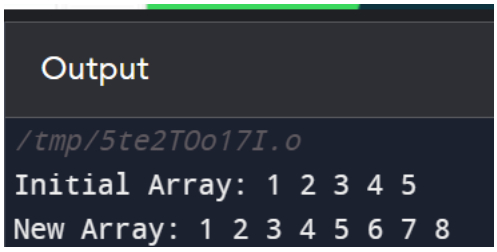
Output

```
/tmp/5te2TOo17I.o
Enter a string: hello
Resultant string after removing duplicates: helo
```

Task 8

```
#include <iostream>
using namespace std;
int main() {
   int a[5] = {1, 2, 3, 4, 5};

   // showing the given initial array
   cout << "Initial Array: ";
   for (int i = 0; i < 5; ++i) {
      cout << a[i] << " ";
   }
   cout << endl;

   // Adding more elements
   int newSize = 8;
   int *newArray = new int[newSize];

   // Copying elements from the old array to the new array
   for (int i = 0; i < 5; ++i) {
      newArray[i] = a[i];
}
   //adding new elements
   newArray[5] = 6;
   newArray[6] = 7;
   newArray[7] = 8;

   // showing the new array
   cout << "New Array: ";
   for (int i = 0; i < newSize; ++i) {
```

```
        cout << newArray[i] << " ";
    }
    cout << endl;
    delete[] newArray;
}
```



Task 9

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

bool findTriplet(int arr[], int n, int X) {
    sort(arr, arr + n);

    for (int i = 0; i < n - 2; ++i) {
        int left = i + 1;
        int right = n - 1;
        while (left < right) {
            int currentSum = arr[i] + arr[left] + arr[right];
            if (currentSum == X) {
```

```
            return true;
        } else if (currentSum < X) {
            ++left;
        } else {
            --right;
        }
      }
   }
   return false;
}

int main() {
   int arr[] = {2, 5, 24, 12, 1, 21};
   int n = sizeof(arr) / sizeof(arr[0]);
   int X = 13;

   if (findTriplet(arr, n, X)) {
      cout << "Triplet found." << endl;
   } else {
      cout << "Triplet not found." << endl;
   }
}
```

Output

```
/tmp/5te2TOo17I.o
Triplet not found.
```

Task 10

```
#include <iostream>
using namespace std;
void swap(int& x, int& y) {
   int temp = x;
   x = y;
   y = temp;
}
void bubbleSort(int arr[], int size) {
   for (int i = 0; i < size - 1; ++i) {
      for (int j = 0; j < size - i - 1; ++j) {
         if (arr[j] > arr[j + 1]) {
            swap(arr[j], arr[j + 1]);
         }
      }
   }
}
int main() {
   const int size = 6;
   int arr[size] = {3, 5, 11, 12, 2, 7};

   cout<< "Original array: ";
   for (int i = 0; i < size; ++i) {
      cout<< arr[i] << " ";
   }

   bubbleSort(arr, size);

   cout<< "\nSorted array: ";
```

```
    for (int i = 0; i < size; ++i) {
        cout<< arr[i] << " ";
    }
}
```