******

***School of Mechanical & Manufacturing Engineering (SMME),***

***National University of Science and Technology (NUST),***

**Fundamentals of Programming**

Program: BE-Aerospace Section: AE-01

Session: Fall 2023 Semester: 1st

Report:

**ASSINGMENT #01**

***Name:*** *Luqman Khalid Bashir*

***CMS:*** 469779

**Question #01:**

Write a C++ program, take two strings 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.

**Code:**

#include <iostream>

using namespace std;

int main()

{

char sent1[30],sent2[30];

int count,counter=0;

cout<<"Enter first string: ";

cin>>sent1;

cout<<"Enter second string: ";

cin>>sent2;

for(int i=0;sent1[i]!='\0'||sent2[i]!='\0';i++)

{ counter++;

    if(sent1[i]!=sent2[i]){++count;}

}

  if (count==0){

  cout<<"Strings are equal."<<endl;

  cout<<"Inverted string is:";

   for(int i=counter;i>=0;i--){

    cout<<sent1[i];}}

else cout<<"Strings are not equal.";

return 0;}

**Output:**

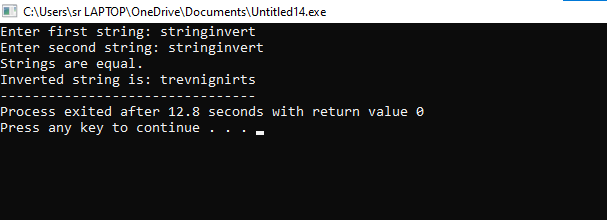


Figure 1:Inverted string

**Question #02:**

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.

#include<iostream>

#include<string.h>

using namespace std;

int main(){

    char string[100];

    cout << "Enter a string.\n";

    cin.getline(string, 100);

    char newstring[100];

    int i=0;

    int k=0;

    while(string[i]!='\0'){

        int check=1;

        for(int j=0;j<k;j++){

            if(newstring[j]==string[i])

            check=0;

        }

        if(check==1){

            newstring[k]=string[i];

            k++;

        }

        i++;

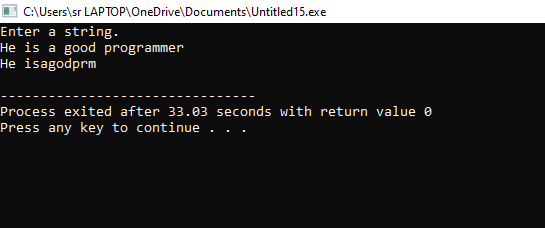
    }

    newstring[k]='\0';

    cout<<newstring<<endl;

    return 0;}

**Output:**

****

**Question #03:**

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

Code:

#include<iostream>

using namespace std;

int main()

{int size;

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

cout<<"How many elements you want to enter:";

cin>>size;

int b[size+5];

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

    b[i]=a[i];

}

cout<<"Enter elements:";

for(int j=5;j<size+5;j++){

    cin>>b[j];

}

cout<<"The new array is: ";

for(int k=0;k<size+5;k++){

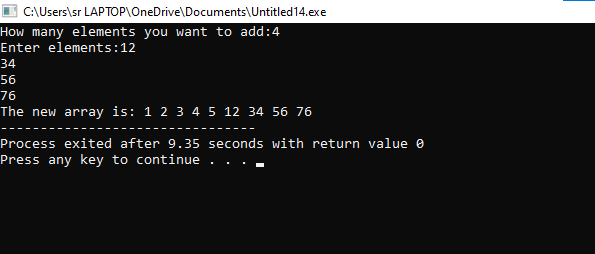
    cout<<b[k]<<" ";

}

    return 0;

}

**Output:**

****

**Figure 2:**Add elements in an array

**Question #04:**

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.

Code:

#include<iostream>

using namespace std;

int main()

{

int n,count=0;

cout<<"Enter number= ";

cin>>n;

int k,i=1;

while(i<=n){

    count=0;

    int j=1;

    while(j<=i){

        if(i%j==0){count++;}

    j++;}

if(count==2){ k=i;}

i++;}

cout<<"The largest prime number equal or less than given number is "<<k;

    return 0;

}

Output:

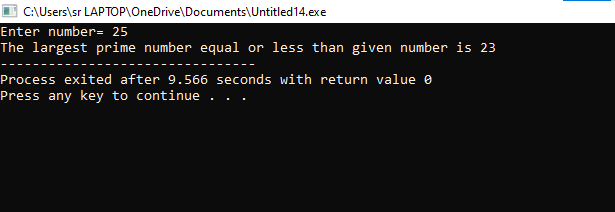


Figure 3:Prime number

**Question #05:**

Implement Bubble Sort on an array of 6 integers.

**Code:**

#include<iostream>

using namespace std;

int main()

{

int array[6];

cout<<"Enter any 6 integers:"<<endl;

for(int i=0;i<6;i++){

    cin>>array[i];

}

cout<<"Unsorted array is: ";

 for (int i= 0; i < 6; i++)

        cout << array[i] << " ";

    cout << endl;

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

for(int j=0;j<5;j++){

    if(array[j]>array[j+1])

{ array[j]=array[j]+array[j+1];

    array[j+1]=array[j]-array[j+1];

    array[j]=array[j]-array[j+1];

   }}

}

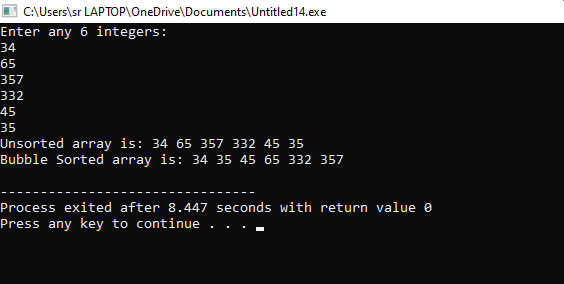
cout<<"Bubble Sorted array is: ";

 for (int i= 0; i < 6; i++)

        cout << array[i] << " ";

    cout << endl;

    return 0;}

****

**Figure 4**:Bubble sort implementation

**Question#06:**

Solve any real-life problem in C++ programming.

#include <iostream>

using namespace std;

double getGradePoints(char grade) {

    switch (grade) {

        case 'A':

        case 'a':

            return 4.0;

        case 'B':

        case 'b':

            return 3.0;

        case 'C':

        case 'c':

            return 2.0;

        case 'D':

        case 'd':

            return 1.0;

        case 'F':

        case 'f':

            return 0.0;

        default:

            return -1.0; // Invalid grade

    }

}

int main() {

    // Get the number of subjects

    int numSubjects;

    cout << "Enter the number of subjects: ";

    cin >> numSubjects;

    // Input grades and credit hours for each subject

    double totalGradePoints = 0.0;

    double totalCreditHours = 0.0;

    for (int i = 0; i < numSubjects; ++i) {

        char grade;

        double creditHours;

        cout << "Enter grade for subject " << i + 1 << ": ";

        cin >> grade;

        cout << "Enter credit hours for subject " << i + 1 << ": ";

        cin >> creditHours;

        double gradePoints = getGradePoints(grade);

        if (gradePoints == -1.0) {

            cout << "Invalid grade entered. Exiting.\n";

            return 1;

        }

        totalGradePoints += gradePoints \* creditHours;

        totalCreditHours += creditHours;

    }

    // Calculate and display CGPA

    if (totalCreditHours > 0.0) {

        double cgpa = totalGradePoints / totalCreditHours;

        cout << "CGPA: " << cgpa << "\n";

    } else {

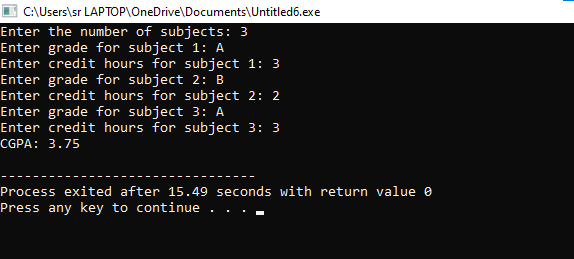
        cout << "Total credit hours is zero. Cannot calculate CGPA.\n";

    }

    return 0;

}

**Output:**

****