PG-DAC September 2023

C++Programming

Assignment -4

Name-Aniket Barsainya

Roll no-230950320010

---

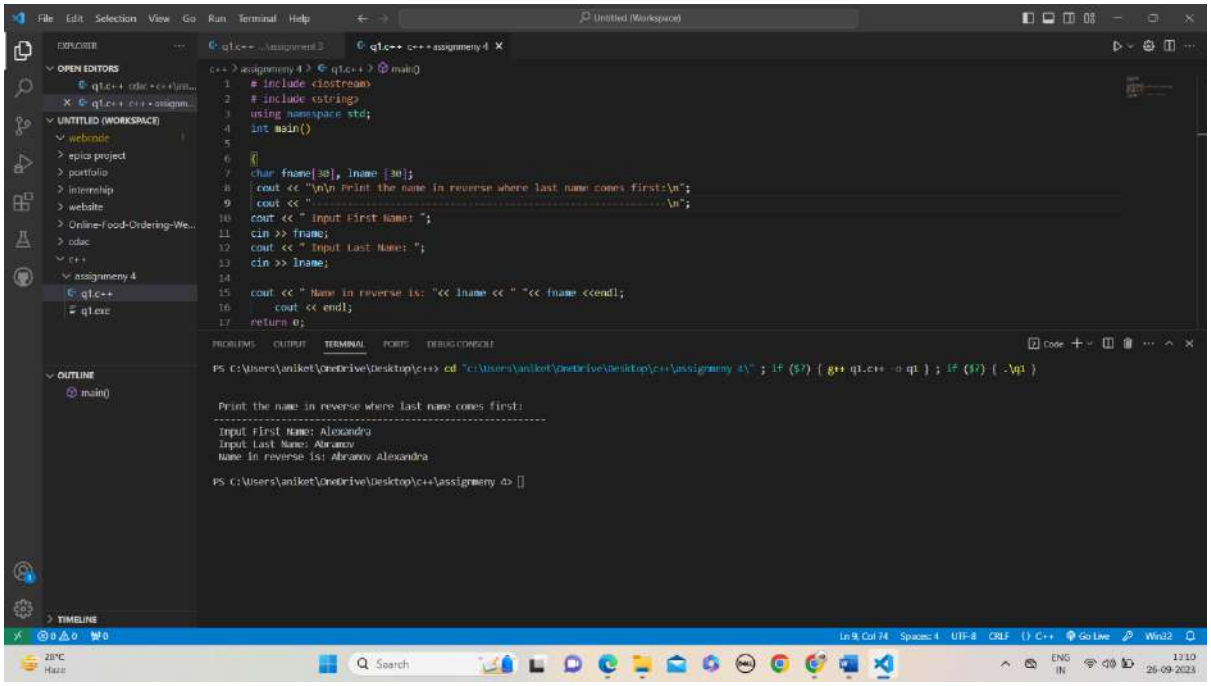
1. Write a C++ program that accepts the user's first and last name and prints them in reverse order with a space between them.
   Sample Output:
   Print the name in reverse where last name comes first:

   Input First Name: Alexandra
   Input Last Name: Abramov
   Name in reverse is: Abramov Alexandra

Code:

```
# include <iostream>
# include <string>
using namespace std;
int main()

{
char fname[30], lname [30];
 cout << "\n\n Print the name in reverse where last name comes first:\n";
 cout << "-----------------------------------------------------------\n";
cout << " Input First Name: ";
cin >> fname;
cout << " Input Last Name: ";
cin >> lname;

cout << " Name in reverse is: "<< lname << " "<< fname <<endl;
    cout << endl;
return 0;
}
```

Output:



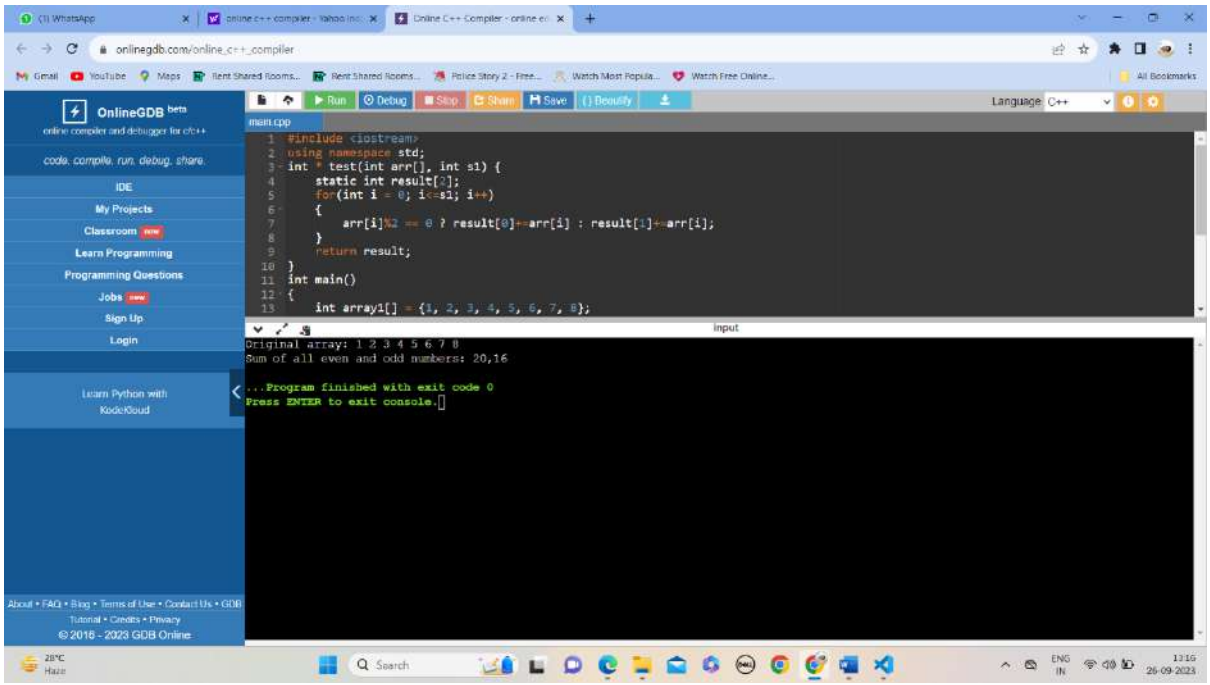
2. Write a C++ program to calculate the sum of all even and odd numbers in an array.
   Sample Output:
   Original array: 1 2 3 4 5 6 7 8
   Sum of all even and odd numbers: 20,16

Code:

```
#include <iostream>
using namespace std;
int * test(int arr[], int s1) {
    static int result[2];
    for(int i = 0; i<=s1; i++)
    {
        arr[i]%2 == 0 ? result[0]+=arr[i] : result[1]+=arr[i];
    }
    return result;
}
int main()
{
    int array1[] = {1, 2, 3, 4, 5, 6, 7, 8};
    int *eo;
    int s1 = sizeof(array1) / sizeof(array1[0]);
    cout << "Original array: ";
    for (int i=0; i < s1; i++)
      cout << array1[i] <<" ";
    eo = test(array1, s1);
```

```
    cout <<"\nSum of all even and odd numbers: " << *(eo+0) << "," << *(eo+1);
    return 0;
}
```

Output:



3. Write a C++ program to print the code (ASCII code / Unicode code etc.) of a given character.
   Sample Output:
   Print code (ASCII code / Unicode code etc.) of a given character:

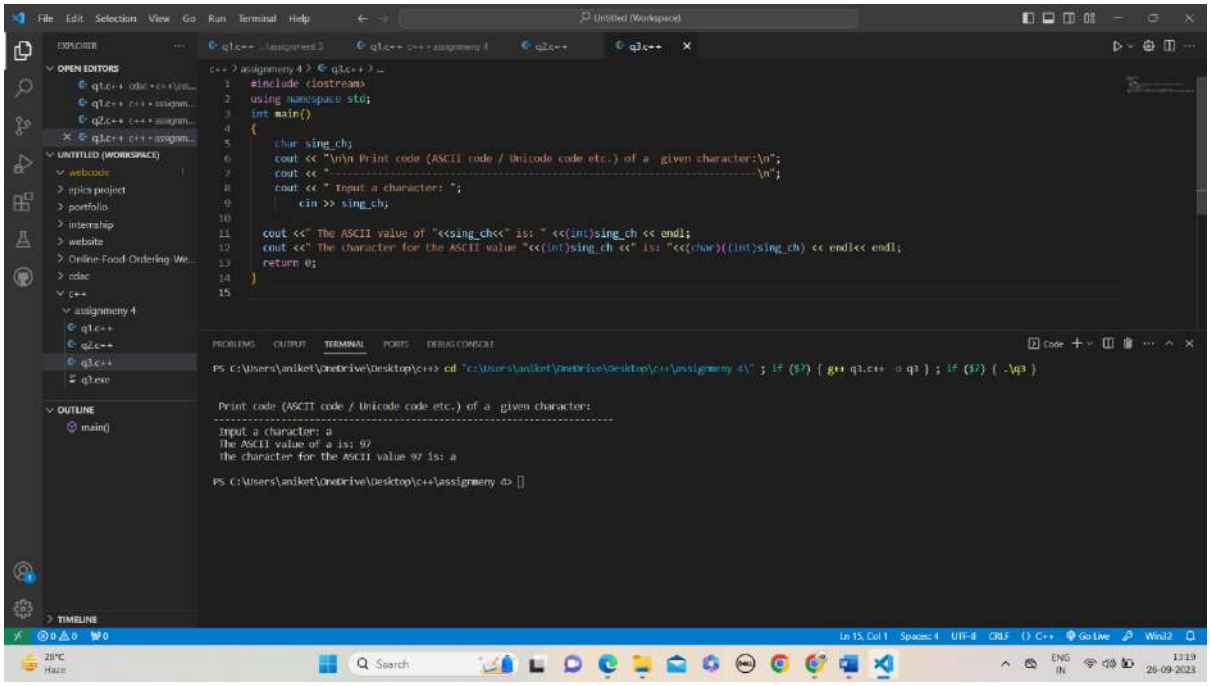
   Input a character: a
   The ASCII value of a is: 97
   The character for the ASCII value 97 is:a

Code:

```
#include <iostream>
using namespace std;
int main()
{
    char sing_ch;
    cout << "\n\n Print code (ASCII code / Unicode code etc.) of a  given character:\n";
```

```
    cout << "----------------------------------------------------------------
------\n";
    cout << " Input a character: ";
        cin >> sing_ch;

  cout <<" The ASCII value of "<<sing_ch<<" is: " <<(int)sing_ch << endl;
  cout <<" The character for the ASCII value "<<(int)sing_ch <<" is:
"<<(char)((int)sing_ch) << endl<< endl;
  return 0;
}
```

Output:



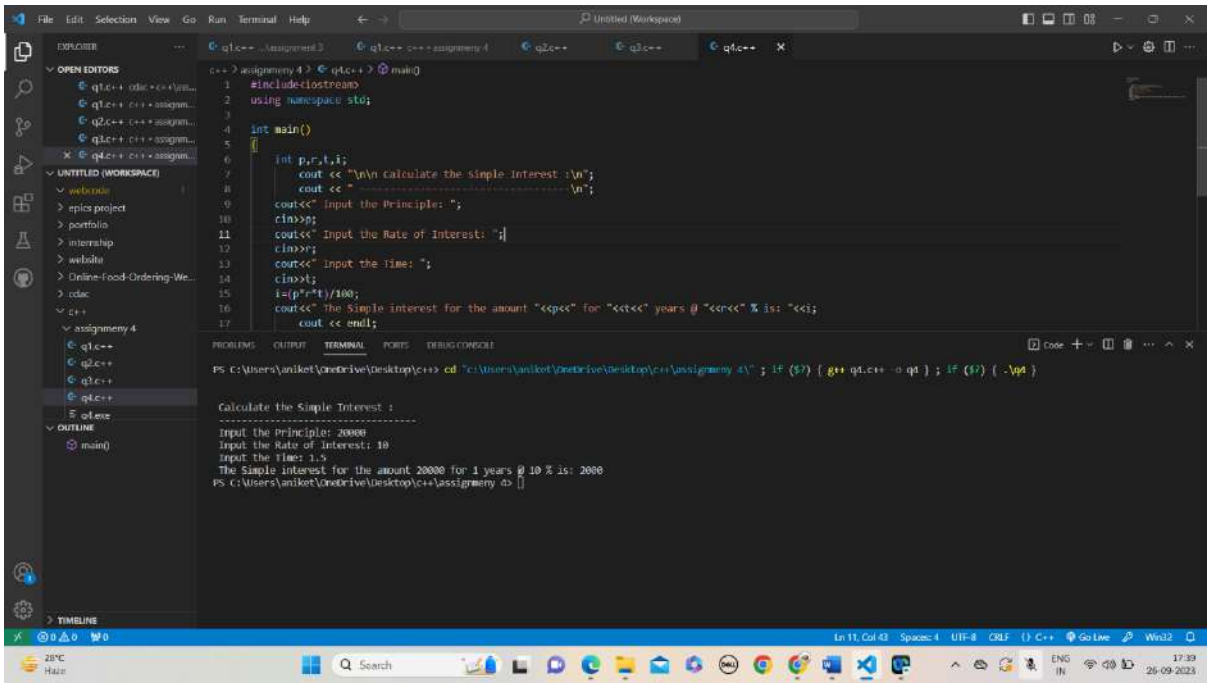
4. Write a C++ program to enter P, T, R and calculate Simple Interest.
   Sample Output:
   Calculate the Simple Interest :

   Input the Principle: 20000
   Input the Rate of Interest: 10
   Input the Time: 1.5
   The Simple interest for the amount 20000 for 1 years @ 10 % is: 2000

Code:

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

int main()
{
    int p,r,t,i;
        cout << "\n\n Calculate the Simple Interest :\n";
        cout << " -----------------------------------\n";
    cout<<" Input the Principle: ";
    cin>>p;
    cout<<" Input the Rate of Interest: ";
    cin>>r;
    cout<<" Input the Time: ";
    cin>>t;
    i=(p*r*t)/100;
    cout<<" The Simple interest for the amount "<<p<<" for "<<t<<" years @
"<<r<<" % is: "<<i;
        cout << endl;
    return 0;
}
```

Output:



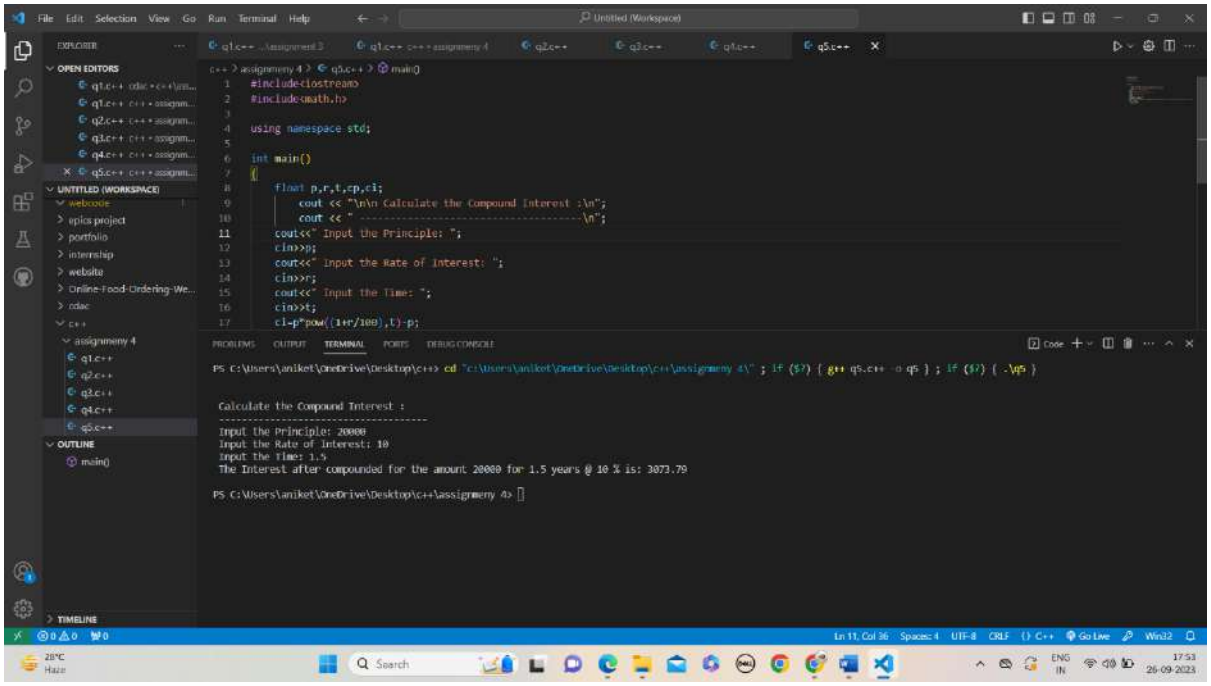
5. Write a C++ program to enter P, T, R and calculate compound interest.
   Sample Output:
   Calculate the Compound Interest :

   Input the Principle: 20000
   Input the Rate of Interest: 10
   Input the Time: 1.5
   The Interest after compounded for the amount 20000 for 1.5 years @ 10% is: 3073.

Code:

```
#include<iostream>
#include<math.h>

using namespace std;

int main()
{
    float p,r,t,cp,ci;
        cout << "\n\n Calculate the Compound Interest :\n";
        cout << " -------------------------------------\n";
    cout<<" Input the Principle: ";
    cin>>p;
    cout<<" Input the Rate of Interest: ";
    cin>>r;
    cout<<" Input the Time: ";
    cin>>t;
    ci=p*pow((1+r/100),t)-p;
    cp=p*pow((1+r/100),t);
    cout<<" The Interest after compounded for the amount "<<p<<" for "<<t<<"
years @ "<<r<<" % is: "<<ci;
    cout << endl;
       cout << endl;
    return 0;
}
```

Output:



6. Write a C++ program to add two binary numbers.
   Sample Output:
   Addition of two binary numbers:

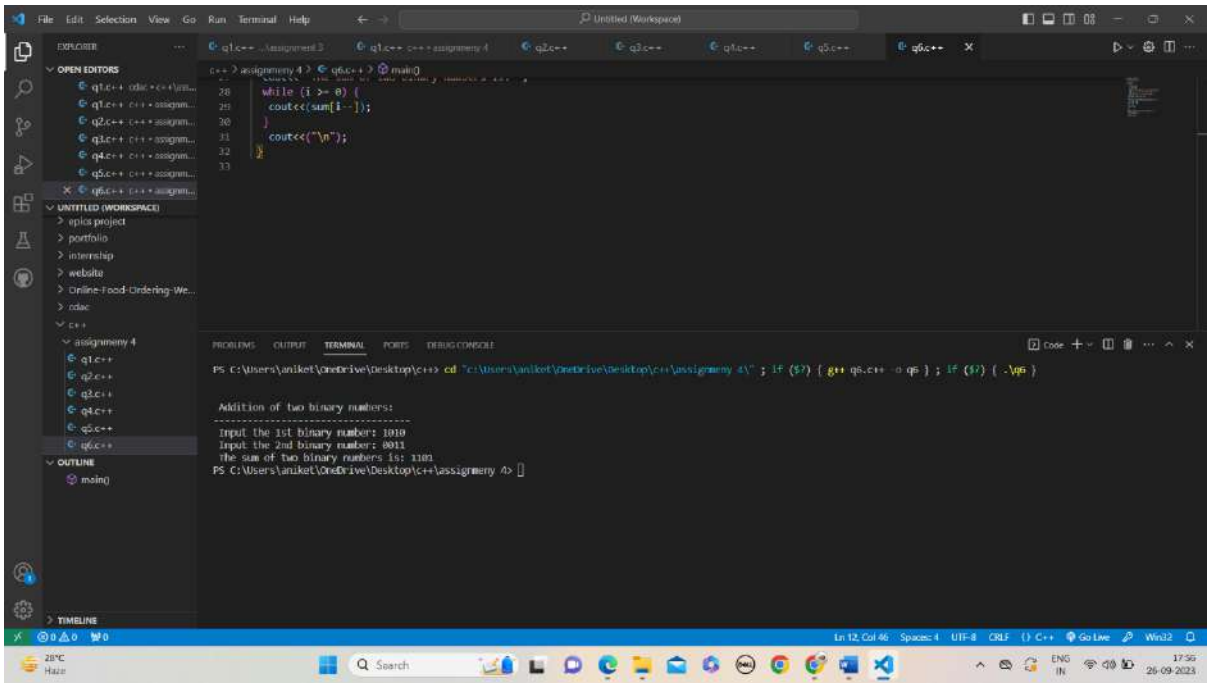
   Input the 1st binary number: 1010
   Input the 2nd binary number: 0011
   The sum of two binary numbers is: 1101

Code:

```
#include <iostream>
#include <math.h>
using namespace std;

int main()
{
    long bn1,bn2;
    int i=0, r=0;
    int sum[20];
    cout << "\n\n Addition of two binary numbers:\n";
    cout << "-----------------------------------\n";
    cout << " Input the 1st binary number: ";
    cin>> bn1;
    cout << " Input the 2nd binary number: ";
```

```
    cin>> bn2;
  while (bn1 != 0 || bn2 != 0)
  {
   sum[i++] = (int)((bn1 % 10 + bn2 % 10 + r) % 2);
   r = (int)((bn1 % 10 + bn2 % 10 + r) / 2);
   bn1 = bn1 / 10;
   bn2 = bn2 / 10;
  }
  if (r != 0) {
   sum[i++] = r;
  }
  --i;
  cout<<" The sum of two binary numbers is: ";
  while (i >= 0) {
   cout<<(sum[i--]);
  }
   cout<<("\n");
 }
```

Output:



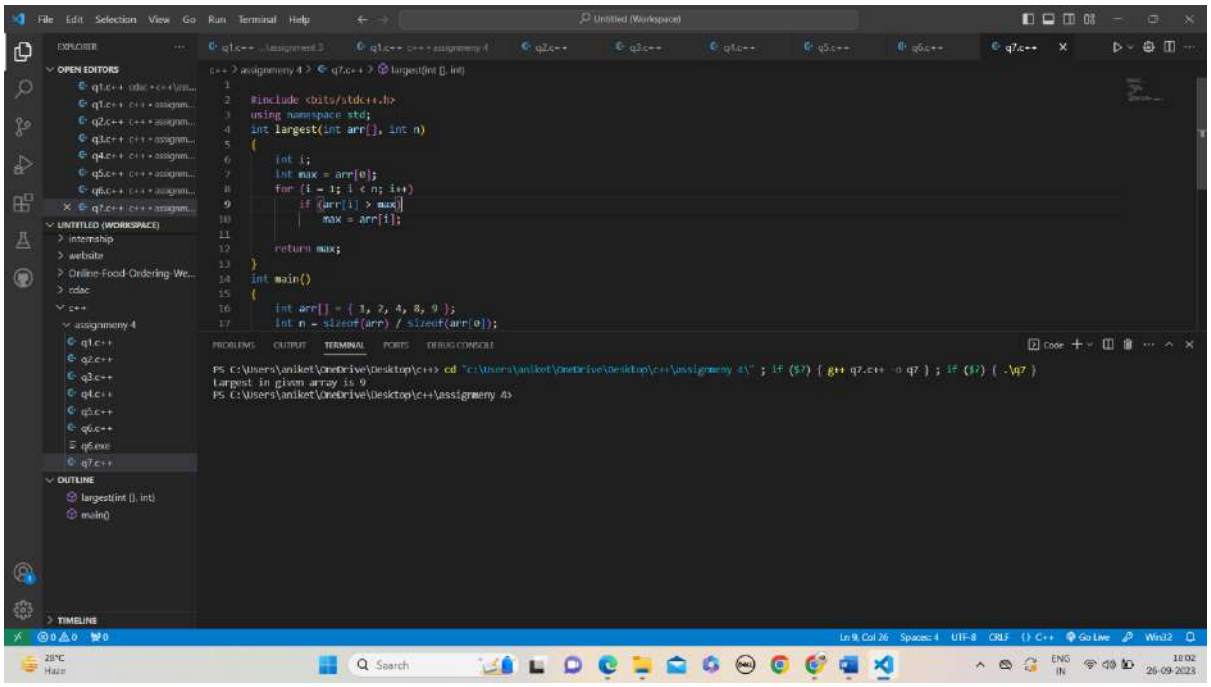
7. Write a C++ program to find the largest element of a given array of integers.

Code:

```
#include <bits/stdc++.h>
```

```cpp
using namespace std;
int largest(int arr[], int n)
{
    int i;
    int max = arr[0];
    for (i = 1; i < n; i++)
        if (arr[i] > max)
            max = arr[i];

    return max;
}
int main()
{
    int arr[] = { 1, 2, 4, 8, 9 };
    int n = sizeof(arr) / sizeof(arr[0]);
    cout << "Largest in given array is " << largest(arr, n);
    return 0;
}
```

Output:

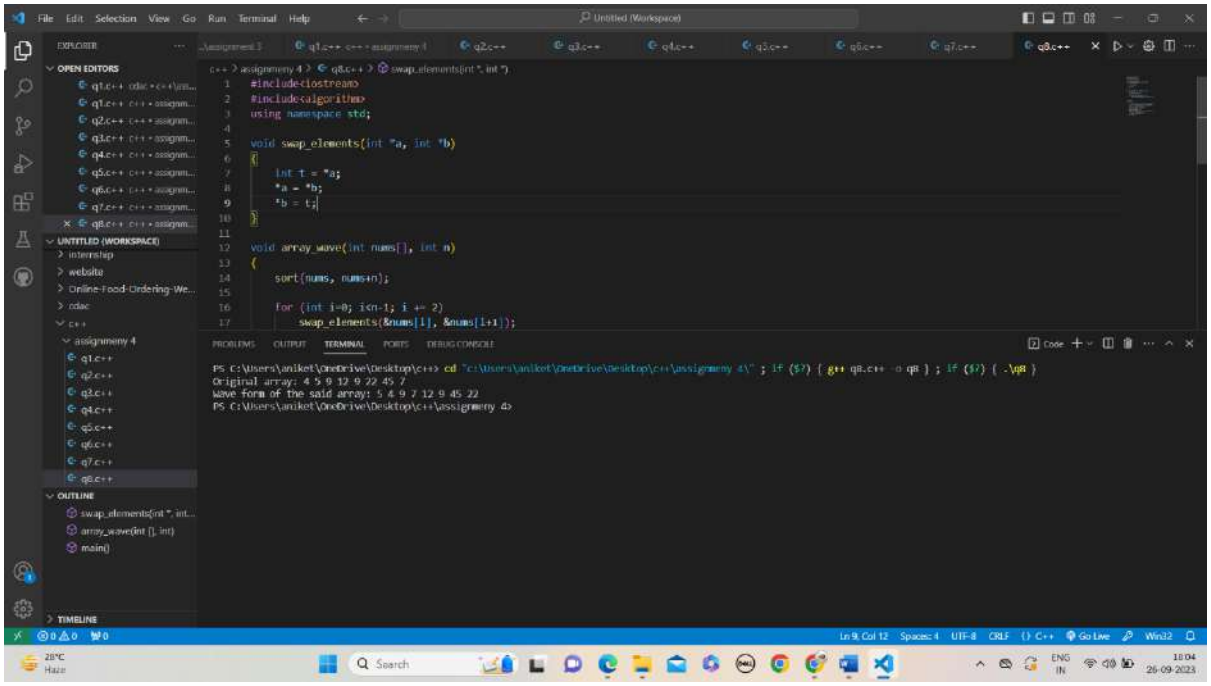


8. Write a C++ program to sort a given unsorted array of integers, in wave form.
   Note: An array is in wave form when array[0] >= array[1] <= array[2] >= array[3] <= array[4] >

Code:

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

void swap_elements(int *a, int *b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

void array_wave(int nums[], int n)
{
    sort(nums, nums+n);

    for (int i=0; i<n-1; i += 2)
        swap_elements(&nums[i], &nums[i+1]);
}

int main()
{
    int nums[] = {4, 5, 9, 12, 9, 22, 45, 7};
    int n = sizeof(nums)/sizeof(nums[0]);
    cout << "Original array: ";
    for (int i=0; i < n; i++)
    cout << nums[i] <<" ";
    array_wave(nums, n);
    cout << "\nWave form of the said array: ";
    for (int i=0; i<n; i++)
       cout << nums[i] << " ";
    return 0;
}
```

Output:



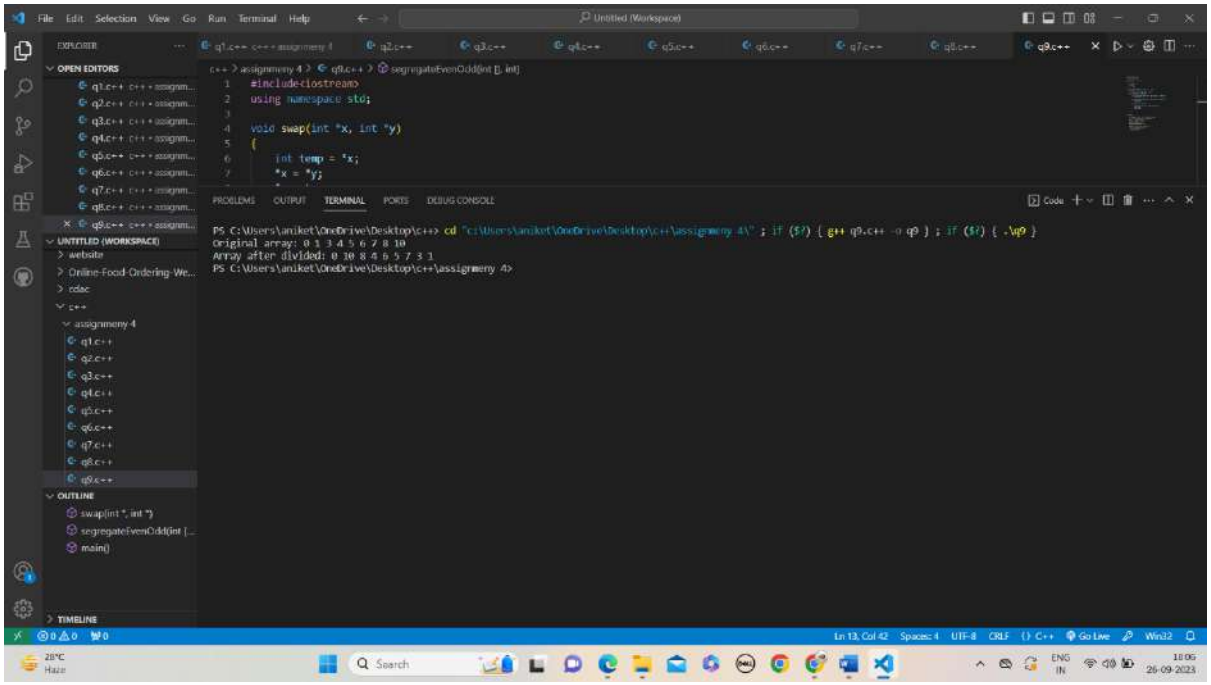
9. Write a C++ program to separate even and odd numbers in an array of integers. Put all even numbers first, and then odd numbers.

Code:

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

void swap(int *x, int *y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

void segregateEvenOdd(int nums[], int size)
{
    int left_num = 0, right_num = size-1;
    while (left_num < right_num)
    {
         while (nums[left_num]%2 == 0 && left_num < right_num)
            left_num++;

        while (nums[right_num]%2 == 1 && left_num < right_num)
            right_num--;
```

```
        if (left_num < right_num)
        {
            swap(&nums[left_num], &nums[right_num]);
            left_num++;
            right_num--;
        }
    }
}

int main()
{
    int nums[] = {0, 1, 3, 4, 5, 6, 7, 8, 10};
    int n = sizeof(nums)/sizeof(nums[0]);
    cout << "Original array: ";
    for (int i=0; i < n; i++)
    cout << nums[i] <<" ";
    segregateEvenOdd(nums, n);

    printf("\nArray after divided: ");
      for (int i=0; i < n; i++)
      cout << nums[i] <<" ";
        return 0;
     }
```

Output:



10. Write a C++ program to find the smallest element of a given array of integers

Code:

```
#include<iostream>
using namespace std;
int main ()
{
    int arr[10], n, i, max, min;
    cout << "Enter the size of the array : ";
    cin >> n;
    cout << "Enter the elements of the array : ";
    for (i = 0; i < n; i++)
        cin >> arr[i];
    min = arr[0];
    for (i = 0; i < n; i++)
    {
        if (min > arr[i])
            min = arr[i];
    }
    cout << "Smallest element : " << min;
    return 0;
}
```

Output: