



Ahsania Mission University of Science & Technology

Department of Computer Science and Engineering

1st Batch, 2nd Year 2nd Semester, Spring 2025

Lab Report

Course Code: CSE 2202

Course Title: Computer Algorithms Sessional

AMUST

Experiment No. : 01
Experiment Date : 5th February, 2025
Submission Date : 12th February, 2025

Submitted To:

Md. Fahim Faisal

Lecturer,

Department of Computer Science and Engineering (CSE)

Faculty of Engineering, Ahsania Mission University of Science & Technology

Submitted By-

Name : -- Md. Aktaruzzaman Aktar --

ID No. : -- 1012320005101015 --

1st Batch, 2nd Year 2nd Semester, Spring 2025

Department of Computer Science and Engineering, AMUST.

Task No.: 01

Problem Statement: Write a program in C/C++ to merge two arrays.

Theory: Merging two 1D arrays involves combining the elements of both arrays into a single array while maintaining their original order.

Here's a step-by-step process for merging two 1D arrays:

- **Create a New Array:** Create a new array that is large enough to hold the combined elements of both arrays. The size of the new array should be the sum of the sizes of the two original arrays.
- **Copy Elements:** Iterate through the elements of the first array and copy them to the new array. Then, iterate through the elements of the second array and copy them to the new array after the elements of the first array.
- **Result:** The new array now contains all the elements from both original arrays, merged in the desired order.

Source Code:

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

    int size1, size2;

    cout << "Enter the size of the first array: ";

    cin >> size1;

    int arr1[size1];

    cout << "Enter " << size1 << " elements for the first array: ";

    for (int i = 0; i < size1; i++) {

        cin >> arr1[i]; // Store the user input in the first array

    }

    cout << "Enter the size of the second array: ";

    cin >> size2;

    int arr2[size2];

    cout << "Enter " << size2 << " elements for the second array: ";

    for (int i = 0; i < size2; i++) {

        cin >> arr2[i];

    }

    int merged[size1 + size2];

    for (int i = 0; i < size1; i++) {

        merged[i] = arr1[i];
```

```

}

for (int i = 0; i < size2; i++) {

    merged[size1 + i] = arr2[i];

}

cout << "The merged array is: ";

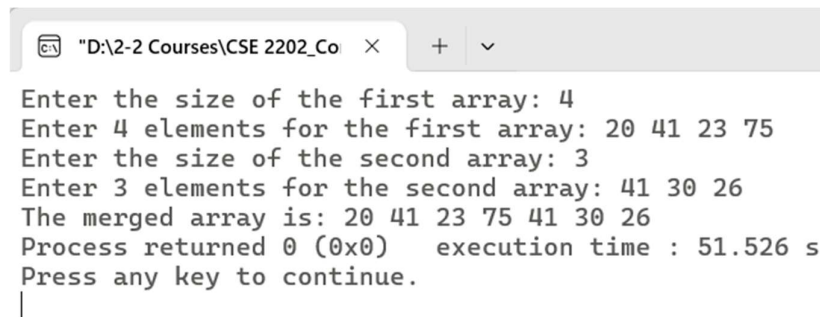
for (int i = 0; i < size1 + size2; i++) {

    cout << merged[i] << " ";

}

```

Output:



The screenshot shows a terminal window titled "D:\2-2 Courses\CSE 2202_Co" with the following output:

```

Enter the size of the first array: 4
Enter 4 elements for the first array: 20 41 23 75
Enter the size of the second array: 3
Enter 3 elements for the second array: 41 30 26
The merged array is: 20 41 23 75 41 30 26
Process returned 0 (0x0)   execution time : 51.526 s
Press any key to continue.
|

```

Conclusion: This program successfully merges two user-defined arrays into a single array. It takes input for two arrays, stores their elements, and then combines them sequentially into a merged array. The implementation demonstrates the use of basic array manipulation and loops in C++. This exercise helps in understanding array operations, user input handling, and memory allocation in C++.

Task No.: 02

Problem Statement: Write a program in C/C++ to find the sum of all the elements of an array.

Source Code:

```

#include <iostream>

using namespace std;

int main() {

    int size;

    cout << "Enter the size of the array: ";

```

```

    cin >> size;

int arr[size];

    cout << "Enter " << size << " elements for the array: ";

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

        cin >> arr[i];

    }

int sum = 0;

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

sum += arr[i];

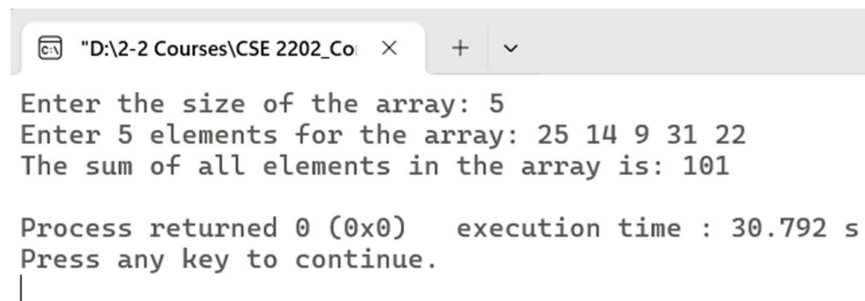
    }

    cout << "The sum of all elements in the array is: " << sum << endl;

}

```

Output:



The screenshot shows a terminal window titled "D:\2-2 Courses\CSE 2202_Co". The output of the program is as follows:

```

Enter the size of the array: 5
Enter 5 elements for the array: 25 14 9 31 22
The sum of all elements in the array is: 101

Process returned 0 (0x0)   execution time : 30.792 s
Press any key to continue.
|

```

Task No.: 03

Problem Statement: Write a program in C/C++ to find maximum in an array.

Source Code:

```

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

    int T;

    cout << "Enter number of test cases: ";

    cin >> T;

```

```

while (T--) {

    int N;

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

    cin >> N;

    int maxHeight = 0;

    cout << "Enter the heights of the mountains: ";

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

        int height;

        cin >> height;

        maxHeight = max(maxHeight, height);

    }

    cout << "Tallest mountain height: " << maxHeight << endl;

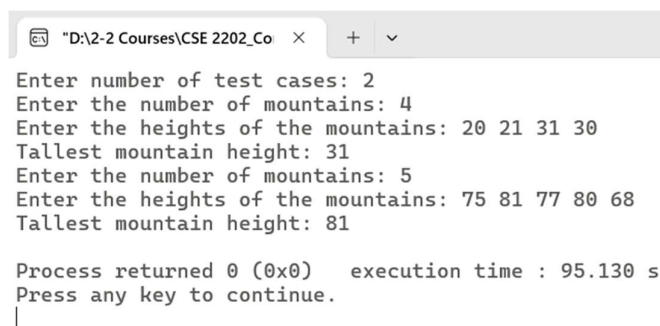
}

return 0;

}

```

Output:



```

"D:\2-2 Courses\CSE 2202_Co  × + ▾
Enter number of test cases: 2
Enter the number of mountains: 4
Enter the heights of the mountains: 20 21 31 30
Tallest mountain height: 31
Enter the number of mountains: 5
Enter the heights of the mountains: 75 81 77 80 68
Tallest mountain height: 81

Process returned 0 (0x0)   execution time : 95.130 s
Press any key to continue.
|

```

Task No.: 04

Problem Statement: Write a program in C/C++ to find MIN to MAX for an array.

Source Code:

```

#include <bits/stdc++.h>

using namespace std;

int main() {

    int t;

```

```

cout << "Enter number of test cases: ";

cin >> t; // Number of test cases

while (t-- > 0) {

    int n;

    cout << "Enter size of the array: ";

    cin >> n; // Size of the array

    vector<int> a(n);

    cout << "Enter " << n << " elements: ";

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

        cin >> a[i]; // Input array elements

    }

    int minValue = *min_element(a.begin(), a.end()); // Find the minimum value

    int count = 0;

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

        if (a[i] != minValue) {

            count++; // Count elements that are not equal to minValue

        }

    }

    cout << "Minimum operations required: " << count << endl; // Output the minimum operations required

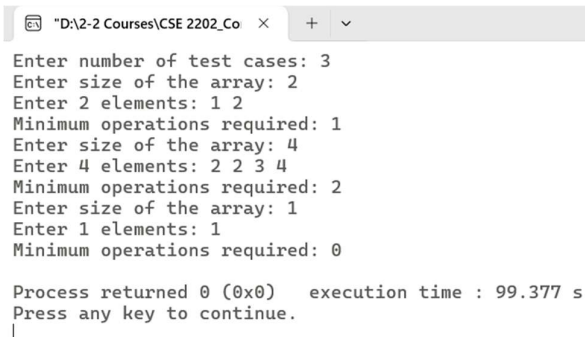
}

return 0;

}

```

Output:



The screenshot shows a Windows command prompt window with the title bar "D:\2-2 Courses\CSE 2202_Co". The program is being executed, and the following input and output are shown:

```

Enter number of test cases: 3
Enter size of the array: 2
Enter 2 elements: 1 2
Minimum operations required: 1
Enter size of the array: 4
Enter 4 elements: 2 2 3 4
Minimum operations required: 2
Enter size of the array: 1
Enter 1 elements: 1
Minimum operations required: 0

Process returned 0 (0x0)   execution time : 99.377 s
Press any key to continue.

```

Task No.: 05

Problem Statement: Write a program in C or C ++ that implements the grade school multiplication algorithm.

Theory: Grade school integer multiplication follows a traditional approach similar to manual long multiplication. Each digit of one number is multiplied by every digit of the other, and the results are aligned based on place value. Carry handling ensures correctness, and the sum of all partial products gives the final result.

Source Code:

```
#include <iostream>

#include <vector>

using namespace std;

void multiply(int A, int B) {

    if (A == 0 || B == 0) {

        cout << "0" << endl;

        return;

    }

    vector<int> result(10, 0);

    int pos = 0, tempB = B;

    while (tempB > 0) {

        int digitB = tempB % 10, carry = 0, tempA = A;

        vector<int> tempRes(10, 0);

        int tempPos = pos;

        while (tempA > 0 || carry) {

            int digitA = tempA % 10;

            int prod = digitA * digitB + carry;

            tempRes[tempPos++] = prod % 10;

            carry = prod / 10;

            tempA /= 10;

        }

    }
```

```

for (int i = 0, carry = 0; i < 10; i++) {

    int sum = result[i] + tempRes[i] + carry;

    result[i] = sum % 10;

    carry = sum / 10;

}

tempB /= 10;

pos++;

}

int i = 9;

while (i > 0 && result[i] == 0) i--;

for (; i >= 0; i--) cout << result[i];

cout << endl;

}

int main() {

    int A, B;

    cout << "Enter two integers: ";

    cin >> A >> B;

    cout << "The product of the two integers using Grade School Algorithm is: ";

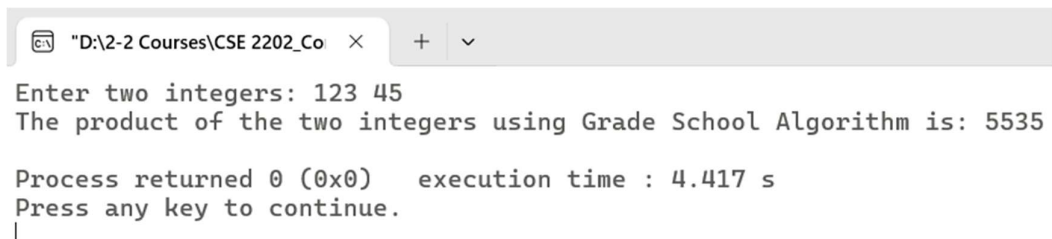
    multiply(A, B);

    return 0;

}

```

Output:



```

D:\2-2 Courses\CSE 2202_Co  X  +  v
Enter two integers: 123 45
The product of the two integers using Grade School Algorithm is: 5535

Process returned 0 (0x0)   execution time : 4.417 s
Press any key to continue.
|

```

Conclusion: This program efficiently multiplies two large integers using digit-by-digit multiplication. It accurately handles carries and correctly formats the output by removing leading zeros. This implementation is useful for scenarios where standard data types cannot store extremely large numbers, such as cryptography and scientific computing.