

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

## ARRAY

### 1) Program to Calculate Average of Numbers Using Arrays

#### CODE:

```
int main()
{
    int n, i;
    float num[100], sum=0.0, average;

    cout << "Enter the numbers of data: ";
    cin >> n;

    while (n > 100 || n <= 0)
    {
        cout << "Error! number should in range of (1 to 100)." << endl;
        cout << "Enter the number again: ";
        cin >> n;
    }

    for(i = 0; i < n; ++i)
    {
        cout << i + 1 << ". Enter number: ";
        cin >> num[i];
        sum += num[i];
    }

    average = sum / n;
    cout << "Average = " << average;

    return 0;
}
```

#### Output

```
Enter the numbers of data: 6
1. Enter number: 45.3
2. Enter number: 67.5
3. Enter number: -45.6
4. Enter number: 20.34
```

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

5. Enter number: 33  
6. Enter number: 45.6  
Average = 27.69

## 2) Program to Find Largest Element of an Array

**CODE:**

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

int main()
{
    int i, n;
    float arr[100];

    cout << "Enter total number of elements(1 to 100): ";
    cin >> n;
    cout << endl;

    // Store number entered by the user
    for(i = 0; i < n; ++i)
    {
        cout << "Enter Number " << i + 1 << " : ";
        cin >> arr[i];
    }

    // Loop to store largest number to arr[0]
    for(i = 1; i < n; ++i)
    {
        // Change < to > if you want to find the smallest element
        if(arr[0] < arr[i])
            arr[0] = arr[i];
    }
    cout << "Largest element = " << arr[0];

    return 0;
}
```

Output

Enter total number of elements: 8

Enter Number 1: 23.4

Enter Number 2: -34.5

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

Enter Number 3: 50  
Enter Number 4: 33.5  
Enter Number 5: 55.5  
Enter Number 6: 43.7  
Enter Number 7: 5.7  
Enter Number 8: -66.5

Largest element = 55.5

### 3) Program to Calculate Standard Deviation

**CODE:**

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

float calculateSD(float data[]);

int main()
{
    int i;
    float data[10];

    cout << "Enter 10 elements: ";
    for(i = 0; i < 10; ++i)
        cin >> data[i];

    cout << endl << "Standard Deviation = " << calculateSD(data);

    return 0;
}

float calculateSD(float data[])
{
    float sum = 0.0, mean, standardDeviation = 0.0;

    int i;

    for(i = 0; i < 10; ++i)
    {
        sum += data[i];
```

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

```
}

mean = sum/10;

for(i = 0; i < 10; ++i)
    standardDeviation += pow(data[i] - mean, 2);

return sqrt(standardDeviation / 10);
}
```

## Output

Enter 10 elements: 1

2

3

4

5

6

7

8

9

10

Standard Deviation = 2.872281

## 4) Program to Access Elements of an Array Using Pointer

### CODE:

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

int main()
{
    int data[5];
    cout << "Enter elements: ";

    for(int i = 0; i < 5; ++i)
        cin >> data[i];

    cout << "You entered: ";
    for(int i = 0; i < 5; ++i)
        cout << endl << *(data + i);

    return 0;
}
```

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

```
}
```

## Output

```
Enter elements: 1
```

```
2
```

```
3
```

```
5
```

```
4
```

```
You entered: 1
```

```
2
```

```
3
```

```
5
```

```
4
```

## 5) Program to Add Two Matrix Using Multi-dimensional Arrays

### CODE:

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int r, c, a[100][100], b[100][100], sum[100][100], i, j;
```

```
    cout << "Enter number of rows (between 1 and 100): ";
```

```
    cin >> r;
```

```
    cout << "Enter number of columns (between 1 and 100): ";
```

```
    cin >> c;
```

```
    cout << endl << "Enter elements of 1st matrix: " << endl;
```

```
    // Storing elements of first matrix entered by user.
```

```
    for(i = 0; i < r; ++i)
```

```
        for(j = 0; j < c; ++j)
```

```
        {
```

```
            cout << "Enter element a" << i + 1 << j + 1 << " : ";
```

```
            cin >> a[i][j];
```

```
        }
```

```
    // Storing elements of second matrix entered by user.
```

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

```
cout << endl << "Enter elements of 2nd matrix: " << endl;
for(i = 0; i < r; ++i)
    for(j = 0; j < c; ++j)
    {
        cout << "Enter element b" << i + 1 << j + 1 << " : ";
        cin >> b[i][j];
    }

// Adding Two matrices
for(i = 0; i < r; ++i)
    for(j = 0; j < c; ++j)
        sum[i][j] = a[i][j] + b[i][j];

// Displaying the resultant sum matrix.
cout << endl << "Sum of two matrix is: " << endl;
for(i = 0; i < r; ++i)
    for(j = 0; j < c; ++j)
    {
        cout << sum[i][j] << " ";
        if(j == c - 1)
            cout << endl;
    }

return 0;
}
```

Output

Enter number of rows (between 1 and 100): 2  
Enter number of columns (between 1 and 100): 2

Enter elements of 1st matrix:  
Enter element a11: -4  
Enter element a12: 5  
Enter element a21: 6  
Enter element a22: 8

Enter elements of 2nd matrix:  
Enter element b11: 3  
Enter element b12: -9  
Enter element b21: 7

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

Enter element b22: 2

Sum of two matrix is:

-1 -4

13 10

## 6) Program to Multiply Two Matrix Using Multi-dimensional Arrays

**CODE:**

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

```
int main()
{
    int a[10][10], b[10][10], mult[10][10], r1, c1, r2, c2, i, j, k;

    cout << "Enter rows and columns for first matrix: ";
    cin >> r1 >> c1;
    cout << "Enter rows and columns for second matrix: ";
    cin >> r2 >> c2;

    // If column of first matrix is not equal to row of second matrix,
    // ask the user to enter the size of matrix again.
    while (c1 != r2)
    {
        cout << "Error! column of first matrix not equal to row of second.";

        cout << "Enter rows and columns for first matrix: ";
        cin >> r1 >> c1;

        cout << "Enter rows and columns for second matrix: ";
        cin >> r2 >> c2;
    }

    // Storing elements of first matrix.
    cout << endl << "Enter elements of matrix 1:" << endl;
    for(i = 0; i < r1; ++i)
        for(j = 0; j < c1; ++j)
        {
            cout << "Enter element a" << i + 1 << j + 1 << " : ";
            cin >> a[i][j];
        }
}
```

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

```
// Storing elements of second matrix.
cout << endl << "Enter elements of matrix 2:" << endl;
for(i = 0; i < r2; ++i)
    for(j = 0; j < c2; ++j)
    {
        cout << "Enter element b" << i + 1 << j + 1 << " : ";
        cin >> b[i][j];
    }

// Initializing elements of matrix mult to 0.
for(i = 0; i < r1; ++i)
    for(j = 0; j < c2; ++j)
    {
        mult[i][j]=0;
    }

// Multiplying matrix a and b and storing in array mult.
for(i = 0; i < r1; ++i)
    for(j = 0; j < c2; ++j)
        for(k = 0; k < c1; ++k)
        {
            mult[i][j] += a[i][k] * b[k][j];
        }

// Displaying the multiplication of two matrix.
cout << endl << "Output Matrix: " << endl;
for(i = 0; i < r1; ++i)
    for(j = 0; j < c2; ++j)
    {
        cout << " " << mult[i][j];
        if(j == c2-1)
            cout << endl;
    }

return 0;
}
```

Output

Enter rows and column for first matrix: 3



# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

2

Enter rows and column for second matrix: 3

2

Error! column of first matrix not equal to row of second.

Enter rows and column for first matrix: 2

3

Enter rows and column for second matrix: 3

2

Enter elements of matrix 1:

Enter elements a11: 3

Enter elements a12: -2

Enter elements a13: 5

Enter elements a21: 3

Enter elements a22: 0

Enter elements a23: 4

Enter elements of matrix 2:

Enter elements b11: 2

Enter elements b12: 3

Enter elements b21: -9

Enter elements b22: 0

Enter elements b31: 0

Enter elements b32: 4

Output Matrix:

24 29

6 25

## 7) Program to Find Transpose of a Matrix

**CODE:**

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

```
int main() {
    int a[10][10], transpose[10][10], row, column, i, j;

    cout << "Enter rows and columns of matrix: ";
    cin >> row >> column;
```

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

```
cout << "\nEnter elements of matrix: " << endl;

// Storing matrix elements
for (int i = 0; i < row; ++i) {
    for (int j = 0; j < column; ++j) {
        cout << "Enter element a" << i + 1 << j + 1 << ": ";
        cin >> a[i][j];
    }
}

// Printing the a matrix
cout << "\nEnter Matrix: " << endl;
for (int i = 0; i < row; ++i) {
    for (int j = 0; j < column; ++j) {
        cout << " " << a[i][j];
        if (j == column - 1)
            cout << endl << endl;
    }
}

// Computing transpose of the matrix
for (int i = 0; i < row; ++i)
    for (int j = 0; j < column; ++j) {
        transpose[j][i] = a[i][j];
    }

// Printing the transpose
cout << "\nTranspose of Matrix: " << endl;
for (int i = 0; i < column; ++i)
    for (int j = 0; j < row; ++j) {
        cout << " " << transpose[i][j];
        if (j == row - 1)
            cout << endl << endl;
    }

return 0;
}
```

Output

Enter rows and columns of matrix: 2

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

3

Enter elements of matrix:

Enter element a11: 1

Enter element a12: 2

Enter element a13: 9

Enter element a21: 0

Enter element a22: 4

Enter element a23: 7

Entered Matrix:

1 2 9

0 4 7

Transpose of Matrix:

1 0

2 4

9 7

## 8) Program to Swap Numbers in Cyclic Order Using Call by Reference

**CODE:**

```
#include<iostream>
```

```
using namespace std;
```

```
void cyclicSwap(int *a, int *b, int *c);
```

```
int main()
```

```
{
```

```
    int a, b, c;
```

```
    cout << "Enter value of a, b and c respectively: ";
```

```
    cin >> a >> b >> c;
```

```
    cout << "Value before swapping: " << endl;
```

```
    cout << "a, b and c respectively are: " << a << ", " << b << ", " << c << endl;
```

```
    cyclicSwap(&a, &b, &c);
```

# SOLUTION BOOK

♥ From SIDDHARTH SINGH

---

```
cout << "Value after swapping numbers in cycle: " << endl;
cout << "a, b and c respectively are: " << a << ", " << b << ", " << c << endl;

return 0;
}

void cyclicSwap(int *a, int *b, int *c)
{
    int temp;
    temp = *b;
    *b = *a;
    *a = *c;
    *c = temp;
}
```

Output

Enter value of a, b and c respectively: 1

2

3

Value before swapping:

a=1

b=2

c=3

Value after swapping numbers in cycle:

a=3

b=1

c=2

**NOTE:** We haven't returned any values from the cyclicSwap() function.