



DINESHRAM A-2212046@nec

0/10 25 0 0 925 2928



Valid Till: 31-May-2026

[Home](#) [Reports](#) [Profile](#) [Help](#) [Logout](#)[Go Back](#)

Completed Programs

Function maximum (Id-13716)

You must define the function **maximum(int a, int b)** which accepts two integers as the input and returns the maximum value between a and b.

Example Input/Output 1:

Input:
15 78

Output:
78

Explanation:
The maximum value between **15** and **78** is **78**.

Example Input/Output 2:

Input:
-5 -10

Output:
-5

[Show My Solution](#)

2212046@nec

28-Aug-2023 14:44:41

DINESHRAM A

2212046

CSE

```
#include <iostream>
using namespace std;
int maximum(int a,int b){
    if(a>b)
        return a;
    else
        return b;
}
int main()
{
    int num1, num2;
    cin >> num1 >> num2;
    cout << maximum(num1, num2);
    return 0;
}
```

Swap Integer Values (Id-13717)

You must define the function **swapIntegerValues** which accepts two integers **a** and **b** as the input. The function must swap the values of a and b.

Note: The modified values of a and b will be printed by the main() function.

Example Input/Output 1:

Input:
89 61

Output:
61 89

Explanation:
Here **a = 89** and **b = 61**.
After swapping the values, **a = 61** and **b = 89**.
Hence the output is 61 89.

Example Input/Output 2:

Input:
159 7531

Output:
7531 159

[Show My Solution](#)

2212046@nec

28-Aug-2023 14:41:47

DINESHRAM A

2212046

CSE

```
#include <iostream>
using namespace std;
void swapIntegerValues(int &a,int &b){
    int t=a;
    a=b;
    b=t;
}
int main()
{
    int a, b;
    cin >> a >> b;
    swapIntegerValues(a, b);
    cout << a << " " << b;
    return 0;
}
```

Rectangles & Count (Id-13908)

Please fill in the missing lines of code to define the class **Rectangle** so that the program prints the following as the output.

10 25 3
15 17 3
20 30 3

[Show My Solution](#)

2212046@nec

28-Aug-2023 14:26:51

DINESHRAM A

2212046

CSE

```
#include <iostream>
using namespace std;
class Rectangle{
public:
    static int rectCount;
    int length,breadth;
    Rectangle(int l,int b){
        length=l;
        breadth=b;
    }
}; int Rectangle :: rectCount=3;

int main()
{
    Rectangle rect1(10, 25);
    Rectangle rect2(15, 17);
    Rectangle rect3(20, 30);
    cout << rect1.length << " " << rect1.breadth << " " << Rectangle::rectCount << endl;
    cout << rect2.length << " " << rect2.breadth << " " << rect2.rectCount << endl;
    cout << rect3.length << " " << rect3.breadth << " " << rect3.rectCount << endl;
    return 0;
}
```

Company - Name & Total Employees (Id-13909)

The program must accept the name and the total employees of a company as the input. The program must print the name and the total employees of the company as shown in the Example Input/Output section. Please fill in the blanks with code to declare and implement the friend function **printDetails** so that the program runs successfully.

Example Input/Output 1:

Input:
Abcde
138

Output:
Company Name: Abcde

Total Employees: 138

Example Input/Output 2:

Input:

KCSTech

1400

Output:

Company Name: KCSTech

Total Employees: 1400

[Show My Solution](#)

2212046@nec

28-Aug-2023 14:38:55

DINESHRAM A

2212046

CSE

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

class Company
{
    string name;
    int totalEmployees;
public:
    Company(string cName, int empCount)
    {
        name = cName;
        totalEmployees = empCount;
    }
    friend void printDetails(Company a);
};
void printDetails(Company a)
{
    cout << "Company Name: " << a.name << endl;
    cout << "Total Employees: " << a.totalEmployees << endl;
}

int main()
{
    string companyName;
    int totalEmployees;
    cin >> companyName >> totalEmployees;
    Company company(companyName, totalEmployees);
    printDetails(company);
    return 0;
}
```

Distance in Feet and Inches (Id-13910)

The program must accept a distance in feet and inches. The program must add 1 inch to the given distance and print the distance as shown in the Example Input/Output section. Please fill in the missing lines of code in the class **Distance** so that the program runs successfully.

Formula: 1 Foot = 12 Inches

Example Input/Output 1:

Input:

5 9

Output:

5 feet 10 inches

Example Input/Output 2:

Input:

6 11

Output:

7 feet 0 inches

[Show My Solution](#)

2212046@nec

28-Aug-2023 14:52:30

DINESHRAM A

2212046

CSE

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

class Distance
{
    int feet;
    int inches;
```

```

public:
Distance (int i,int j){
    feet=i;
    inches=j;
}
friend void addOneInch(Distance &d);
friend void printDistance(Distance d);
};

void addOneInch(Distance &d)
{
    if(d.inches == 11)
    {
        d.feet++;
        d.inches = 0;
    }
    else
    {
        d.inches++;
    }
}

void printDistance(Distance d)
{
    cout << d.feet << " feet " << d.inches << " inches" << endl;
}

int main()
{
    int feet, inches;
    cin >> feet >> inches;
    Distance distance(feet, inches);
    addOneInch(distance);
    printDistance(distance);
    return 0;
}

```

Matrix Vector Multiplication (Id-13911)

The program must accept a matrix of size **3*4** and a vector of size **4** as the input. The program must print the product of the matrix and the vector as the output. Please fill in the missing lines of code so the program runs successfully.

Example Input/Output 1:

Input:

```

1 2 3 4
2 4 6 8
9 5 7 3
10 20 30 40

```

Output:

```

300
600
520

```

Explanation:

The product of the matrix and the vector are given below

300 -> ((1*10)+(2*20)+(3*30)+(4*40))

600 -> ((2*10)+(4*20)+(6*30)+(8*40))

520 -> ((9*10)+(5*20)+(7*30)+(3*40))

Example Input/Output 2:

Input:

```

2 4 6 8
1 3 5 7
5 6 7 8
100 200 300 400

```

Output:

```

6000
5000
7000

```

Show My Solution

2212046@nec

28-Aug-2023 15:18:15

DINESHRAM A

2212046

CSE

```

#include <iostream>
#define ROWS 3

```

```
#define COLS 4

using namespace std;

class Vector;

class Matrix
{
public:
    int R,C;
    int matrix[ROWS][COLS];
    Matrix()
    {
        R=ROWS;
        C=COLS;
    }

    void storeValue(int val,int row,int col)
    {
        matrix[row][col]=val;
    }

    friend void multiply(Matrix&, Vector&);
};

class Vector{
public:
    int index;
    int vector[COLS];
    void storeValue(int x,int i){
        vector[i]=x;
    }friend void multiply(Matrix&,Vector&);
};

void multiply(Matrix& a,Vector& b){
    for(int i=0;i<ROWS;i++){
        int res=0;
        for(int j=0;j<COLS;j++){
            res+=((a.matrix[i][j])*(b.vector[j]));
        }cout<<res<<endl;
    }
}

int main()
{
    int currValue;
    Matrix matrix;
    for(int row=0; row<ROWS; row++)
    {
        for(int col=0; col<COLS; col++)
        {
            cin >> currValue;
            matrix.storeValue(currValue,row,col);
        }
    }

    Vector vect;
    for(int index = 0; index < COLS; index++)
    {
        cin >> currValue;
        vect.storeValue(currValue, index);
    }

    multiply(matrix,vect);
    return 0;
}
```