



DINESHRAM A-2212046@nec



0/10



25



0



0



966



2737



Valid Till:

31-May-2026



Home



Reports



Profile



Help



Logout

Go Back

Completed Programs

Add Two Distances (Id-13672)

The function/method **add** accepts two arguments **d1** and **d2** representing the **objects** of the class **Distance**.

The function/method must return a new object of the class **Distance** representing the sum of the given two distances.

Your task is to define the function **add** so that the program runs successfully.

Example Input/Output 1:

Input:

3 10

2 11

Output:

3 10

2 11

6 9

Explanation:

Distance 1 = 3 feet and 10 inches.

Distance 2 = 2 feet and 11 inches.

The sum of the two distances is **6 feet and 9 inches**.

Example Input/Output 2:

Input:

4 5

4 7

Output:

4 5

4 7

90

[Show My Solution](#)

2212046@nec

25-Sep-2023 14:26:28

DINESHRAM A

2212046

CSE

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

class Distance
{
public:
    int feet;
    int inches;
    Distance(int dFeet, int dInches)
    {
        feet = dFeet;
        inches = dInches;
    }
};

Distance add(Distance &a, Distance &b){
    Distance x(0,0);
    x.feet=a.feet+b.feet;
    x.inches=(a.inches+b.inches);
    if(x.inches>=12){
        x.feet++;
        x.inches-=12;
    }return x;
}

int main()
{
    int feet, inches;
    cin >> feet >> inches;
    Distance distance1(feet, inches);
    cin >> feet >> inches;
    Distance distance2(feet, inches);
    Distance distance3 = add(distance1, distance2);
    cout << distance1.feet << " " << distance1.inches << endl;
    cout << distance2.feet << " " << distance2.inches << endl;
    cout << distance3.feet << " " << distance3.inches << endl;
    return 0;
}
```

Function removeUnitDigits (Id-13860)

You must define the function **removeUnitDigits(int arr[], int N)** which accepts an integer array **arr** and it's size **N** as the input. The function must remove the unit digits of the integers in the array **arr** and store the results in a new array. Then the function must return the new array.

Note: The main() function will print the integers in the array returned by the function removeUnitDigits(int arr[], int N).

Input Format:

The first line contains N.

The second line contains N integers separated by a space.

Output Format:

The first line contains N integers separated by a space.

Example Input/Output 1:

Input:

5
522 85 604 648 55

Output:

52 8 60 64 5

Explanation:

After removing the unit digits in the given **5** integers, the integers become **52, 8, 60, 64** and **5**. So they are printed as the output.

Example Input/Output 2:

Input:

8
66 5650 938 72 7294 929 12 9639

Output:

6 565 93 7 729 92 1 963

[Show My Solution](#)

2212046@nec

27-Sep-2023 14:02:12

DINESHRAM A

2212046

CSE

```
#include <iostream>
using namespace std;
int *removeUnitDigits(int arr[],int N){
    int *ptr=new int[N];
    for(int i=0;i<N;i++){
        ptr[i]=arr[i]/10;
    }return ptr;
};
int main()
{
    int N;
    cin >> N;
    int arr[N];
    for(int index = 0; index < N; index++)
    {
        cin >> arr[index];
    }
    int *ptr = removeUnitDigits(arr, N);
    for(int index = 0; index < N; index++)
    {
        cout << ptr[index] << " ";
    }
}
```

```
    }  
    delete []ptr;  
    return 0;  
}
```

Total Marks - N Students (Id-13861)

The program must accept the marks of **N** students as the input. The program must print the total marks of the N students as the output. Please fill in the missing lines of code so that the program runs successfully.

Input Format:

The first line contains N.

The second line contains N integers representing the marks of the N students.

Output Format:

The first line contains the total marks of the N students.

Example Input/Output 1:

Input:

4
50 95 78 80

Output:

303

Example Input/Output 2:

Input:

5
48 59 26 42 86

Output:

261

Show My Solution

2212046@nec

27-Sep-2023 14:08:57

DINESHRAM A

2212046

CSE

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

```
class Student  
{  
    int marks;  
public:  
    Student(int m)  
    {  
        marks = m;  
    }  
    int getMarks()  
}
```

```
    {
        return marks;
    }
};
Student **getStudents(int n){
    int marks[n];
    Student *p=(Student *)malloc(sizeof(Student)*n);
    Student **o=(Student **)malloc(sizeof(Student)*n);
    for(int i=0;i<n;i++){
        cin>>marks[i];
        *(p+i)=marks[i];
        *(o+i)=&p[i];
    }return o;
}
int main()
{
    int N;
    cin >> N;
    Student** stud = getStudents(N);
    int totalMarks = 0;
    for(int index = 0; index < N; index++)
    {
        totalMarks += stud[index]->getMarks();
    }
    cout << totalMarks;
    return 0;
}
```