Queens College, CUNY, Department of Computer Science
Instructor: Dr. Sateesh Mane


## Question

- Write a function to return the sum of the elements of an array $a$ of length $n$:

$$\text{sum} = \sum_{i=1}^{n} a_i \, .$$

- The problem was assigned to three classes, with array types `int`, `long` or `double`, respectively.
- **Write two functions, one in C++ and the other in Java.**

## Student solutions

- **Students in all my classes are expected to write program code.**
- **Students in my elective/senior level classes are expected to be competent programmers.**
- Student solutions are displayed below.
- The solutions are ranked in order of decreasing quality.
- **Write your own solution and find where do you rank in the list below.**
- Note the following:
  1. The name of the function was not specified above.
  2. The function names in the student solutions were changed to one of the following:
     (a) `sum_int` for array of type `int`.
     (b) `sum_long` for array of type `long`.
     (c) `sum_double` for array of type `double`.
- *I forgive minor errors and typos.* The student code (with typo, etc.) is commented out and my substitution is written on the line below.
- **Major errors are *not* forgiven.**

# Correct solutions

**Correct solutions are not displayed.**
**This document is not for you to copy answers from those who know what they are doing.**

## Off by one

- Solutions where the array indices were off by one are not displayed below.

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

- I forgive such errors as a typo which can be easily fixed.
- **Nevertheless, it is a mistake you should not make.**
  1. In Java the above loop would throw an "array out of bounds" exception.
  2. In C++ the above loop would result in undefined behavior: dangerous.

## Missing arguments

- Fundamentally correct solutions where some inputs are missing are not displayed below.

```
int student_func(int a[])
{
  ...
  for (int i = 0; i < n; i++)
  ...
}
```

- I forgive this as a "writing quickly under time pressure" error.
- The compiler would immediately identify the mistake and it is easily fixed.

```
// unreliable, call function twice with same input array
long s = 0;

//long sum_long(s, n, a[]) {
long sum_long(long s, int n, long a[]) {
  for (int i = 0; i < n; i++) {
    s = s + a[i];
  }
  return s;
}
```

```
// unreliable, call function twice with same input array
public static long s = 0;

//long sum_long(s, n, a[]) {
public static long sum_long(long s, int n, long a[]) {
  for (int i = 0; i < n; i++) {
    s = s + a[i];
  }
  return s;
}
```

```
int sum_long(int n) {
  int sum = 0;
  long a[n] = 0;
  for (i = 0; i < sizeof(a); i++) {
    sum = sum + a[i];
  }
  return sum;
}
```

```
public static int sum_long(int n) {
  int sum = 0;
  long a[n] = 0;
  for (i = 0; i < a.length; i++) {
    sum = sum + a[i];
  }
  return sum;
}
```

C++ code not compile

```
//public static double sum_double(arr[]) {
public static double sum_double(double arr[]) {
    int i = 0;
    int sum = 0;
    //while (i <= arr.length()) {
    while (i <= arr.length) {
        sum = sum + arr[i];
    }
    return sum;
}
```

```
//double sumArray(double[] array)
double sum_double(double array[])
{
  double sum = 0.0;
  for (int i = 1; i >= 1; i++)
    sum += array[i];
  return sum;
}
```

Java code works

```
//static void arraySum() {
void sum_double() {
  double sum = 0.0;
  double d[n];
  for (int i = 0; i < n; i++) {
    sum = sum + d[i];
  }
  cout << d[i];
}
```

```
//public static void arraSum () {
public static void sum_double() {
  double sum = 0.0;
  double d[] = new double[n];
  for (int i = 0; i < n; i++) {
    sum = sum + d[i];
    System.out.print(" " + d[i];
  }
}
```

```
int sum = 0;
int n = ?;  <-- choose n
for (int i = 1; i <= n; i++) {
  sum += i;
 }
cout << "sum of 1 through " + n + " is " + sum;
```

```
// Java code with loop
    int sum = 0;
    int n = ?;   // <-- choose value for n
    for (int i = 1; i <= n; i++) {
        sum += i;
    }
    System.out.println("sum of 1 through " + n + " is " + sum);

// Java code without loop
    int sum;
    int n = ?;   // <-- choose value for n
    sum = n*(n+1)/2;
    System.out.println("sum of 1 through " + n + " is " + sum);
```

C++ not do

```
public double sum(double[] myarray) {
    double sum = 0;
    for (int i = 0; i++; i < myarray.length) {     // read this carefully
        sum = sum + i;
    }
    return sum;
}
```

```
int sum = 0; int x;
int main() {
  for (int x = 1; x <= n; x++)
    sum += x;
  cout << sum << endl;
  return 0;
}
```

```
      public static void main(String[] args) {
        int sum = 0; int x;
        for (int x = 1; x <= n; x++)
          sum += x;
        System.out("The sum is " + sum);
      }
```

```
int main() {
  int n, sum = 0;
  cout << "Enter a positive integer: ";
  cin >> n;
  for (int i = 1; i <= n; ++i) {
    sum += i;
  }
  cout << "sum = " << sum;
  return 0;
}
```

```
    public static void main(String[] args) {
        int sum = 0;
        int n;
        System.out.println("Enter a positive integer: ");
        Scanner s = new Scanner();
        n = s;
        for (int i = 1; i <= n; ++i) {
            sum = sum + i;
        }
        System.out.println(sum);
    }
```

```
int main { int n;
  int x[n];
  int i;
  for (i = 0; i < 0; i++)
    int sum = sum + x[i];
  for (i = 0; i < n; i++)
    cout << sum;
}
```

```
    public static void main
        int n, sum = 0;
    Snanner s = new Scanner(System.in);
    for (int i = 0; i < n; i++) {
        s[i] = s.nextln();
        sum = sum + a[i];
    }
    System.out.println("sum: sum);
}
}
```

```
int sum;
for (i = 1; i < j; i++)
  {
    size.x = j;
    sum = 0;
    sum = sum + x[i];
  }
return sum;
```

```
    int someArray[];
    int sum = 0;
    for (int i : someArray) {
        sum += i;
    }
    System.out.println(sum);
```

```
int sum = 0;
x[n] = {    };
//int sum(int[] arr, int k) {
int sum_int(int arr[], int k) {
  for (int i = 1; i < k-1; i++) {
    sum = arr[i] + sum;
  }
  //cout sum;
  cout << sum;

  // what is the return value?
}
```

```
    //public static int sum(int[] arr, int k) {
    public static int sum_int(int[] arr, int k) {
        for (int i = 1; i < k-1; i++) {
            sum = arr[i] + sum; }
        System.out.println(sum);

        // what is the return value?
    }
```

```
void sum_int(int n) {
  for (int i = 0; i < n; i++) {
    cin >> arr[i];
    cout << "sum of elements";
  }
  for (int i = 0; i < n; i++) {
    sum = sum + arr[i];
  }
  for (int i = 0; i < n; i++)
    cout << sum;
}
```

```
    public static int sum_int(int n) {
        int c = 0;
        if (a[n] == 0) c = 0;
        for (int i = 0; i > a.length(); i++) {
            c = n + a.length[i];
        }
        return c;
    }
```

C++ not do

```
    int sum = 0;
    for (i = 0; i < n; i++) {
        sum += i;
    }
    System.out.println("The sum from 1 through +n + is : + sum + ");
```

C++ code works

```
    //void sumfun(double array[], int n) {
    public static double sum_double(double array[], int n) {
        int = n;
        Scanner obj = new Scanner(System.in);
        d = obj.nextln();
        for (int i = 0; i < n; i++) {
            a[i] = obj.nextln();
            sum = sum + array[i];
        }
        System.out.println(sum);
    }
```

```
//int sum_int(int n);      <--- note the semicolon
int sum_int(int n) {
  for (int i = 1; i < n; i++) sum += i;
  return sum;
}
```

Java not do

```
int sum_int(int x)
  int n = 10;
int x[i] = {   };
for (int i = 0; i <= n; i++) ;
int sum = x[i]++;
return sum;
```

```
int sum_int(int x)
  int x = 10;
int x[i] = {   };
for (int i = 0; i <= n; i++) ;
    int sum = intstream.of(x).sum();
return sum;
```

C++ almost satisfactory

```
int n, sum = 0;
Scanner s = new Scanner(System.in);
n = s.nextInt();
double a[] = new int(n);
for (int i = 0; i < n; i++)
{
    a[i] = s.nextInt();
    sum = sum + a[i];
}
System.out.println(sum);
```

```
double sum_double(int n) {
  x = 0;
  for (int i=0; i <= n.length; i++) {
    return x*x[i];
  }
}
```

```
    public static double sum_double(int n) {
        double x = 0;
        for (int i=0; i < n.length; i++) {
            x *= x[i];
        }
        return x;
    }
```

```
//public arraySum(n) {
double sum_double(int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
      array[i] + sum;
      cout << sum;
    }
}
```

```
//public arraySum(n) {
public static double sum_double(int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) { array[i] + sum;
        return sum;
    }
}
```

C++ not do

```
double sum = 0.0;
int n = 50;
double[] arrayOne = new double[];
Scanner keyboard = new Scanner[Syatem.in];

//public double anArray(double[] n)
static double sum_double(double [] a)
{
    for (int i = 0; i < anArray.length; i++)
        {
            System.out.println("Enter value for " + (i+1));
            anArray[i] = Keyboard.nextDouble();
            sum = sum + anArray[i];
        }
    return sum;
}
```

C++ not do

```
//public double sum_double(int n) {
public static double sum_double(int n) {
    double addends[] = new double[n];
    double result = 0;
    for (int i = 0; i < n; i++) {
        addends[i] = i++;
    }
    for (int i = 0; i < n; i++) {
        result += addends[i];
    }
    return result;
}
```

```
void sum() {
  d[n];
  int length = d.size();
  for (int i = 1; i = n; i++) {
    double sum = sum + d[i]; }
  cout >> sum;
}
```

```
public class sum() {
    double[] d;
    int length = d.size();
    for (int i = 1; i = n; i++) {
        double sum = sum + d[i]; }
    System.out.println(sum);
}
```

```
public double sigma(double n) {
  if (n >= 1) {
    cout << "Enter a value for i " << endl;
    double i = 0.0;
    cin >> i;
  }
}

int main() {
  sigma(1.0);
  double sum = 0.0;
  for (int i = 1; i <= n; i++) {
    sum = sum + i;
  }
  return sum;
}
```

```
public double sigma(double n) {
    if (n >= 1) {
        System.out.println("Enter a value for i ");
        input = input.nextDouble();
    }
}

public static void main(String[] args) {
    sigma(1.0);
    double sum = 0.0;
    for (int i = 1; i <= n; i++) {
        sum = sum + i;
    }
    return sum;
}
```

```
// unreliable, call function twice with same input
static long sum = 0;
static int n = 4;
long sum_long(long a[])
{
  for (int i = 0; i < n; i++) {
    sum += a[i];
  }
  return sum;
}
```

```
    // unreliable, call function twice with same input array
    static long sum = 0;
    static int n = 4;
    public static long sum_long(long []a)
    {
        for (int i = 0; i < n; i++)
            {
                sum += a[i];
            }
        return sum;
    }
```

```
    // updated student solution, this is even worse
    static long sum=0;
    static int n=3;
    static boolean once;                        // is this initialized properly?
    public static long sum_long(long[] a) {
        if(!once){
            for(int i=0;i<n;i++){
                sum+=a[i];
            }
            once=true;
        }

        return sum;
    }
```

```
C++ not do
```

```
function sum
int x = 0
    for (int = i; i != 0; i++)
    for (int = j; j != 0; j++)
        a[i][j] += x;
```

```
class mySumArray {
  int m = n;
  long myArray* calcTheArraySum;
  sumArray(int n)
  int m = n;
  calcTheArraySum[m];
  calcTheArraySum = new calcArraySum;
  for (int i = 0; i < n; i++) {
    theArraySum[i] = new myArray[i];
  }
}
```

```
Java not do
```