**package** Question21;

**class** FindTriplet

{

**boolean** find3Numbers(**int** A[], **int** arr\_size, **int** sum)

{

**int** l, r;

quickSort(A, 0, arr\_size - 1);

**for** (**int** i = 0; i < arr\_size - 2; i++)

{

l = i + 1;

r = arr\_size - 1;

**while** (l < r)

{

**if** (A[i] + A[l] + A[r] == sum)

{

System.***out***.print("Triplet is " + A[i] + " ," + A[l]

+ " ," + A[r]);

**return** **true**;

}

**else** **if** (A[i] + A[l] + A[r] < sum)

l++;

**else** // A[i] + A[l] + A[r] > sum

r--;

}

}

// If we reach here, then no triplet was found

**return** **false**;

}

**int** partition(**int** A[], **int** si, **int** ei)

{

**int** x = A[ei];

**int** i = (si - 1);

**int** j;

**for** (j = si; j <= ei - 1; j++)

{

**if** (A[j] <= x)

{

i++;

**int** temp = A[i];

A[i] = A[j];

A[j] = temp;

}

}

**int** temp = A[i + 1];

A[i + 1] = A[ei];

A[ei] = temp;

**return** (i + 1);

}

**void** quickSort(**int** A[], **int** si, **int** ei)

{

**int** pi;

**if** (si < ei)

{

pi = partition(A, si, ei);

quickSort(A, si, pi - 1);

quickSort(A, pi + 1, ei);

}

}

**public** **static** **void** main(String[] args)

{

FindTriplet triplet = **new** FindTriplet();

**int** A[] = {1, 4, 45, 6, 10, 8};

**int** sum = 22;

**int** arr\_size = A.length;

triplet.find3Numbers(A, arr\_size, sum);

}

}