**Attending Workshops**

A student signed up for  workshops and wants to attend the maximum number of workshops where no two workshops overlap. You must do the following:

Implement  [structures](http://www.cplusplus.com/doc/tutorial/structures/):

1. *struct Workshop* having the following members:
   * The workshop's start time.
   * The workshop's duration.
   * The workshop's end time.
2. *struct Available\_Workshops* having the following members:
   * An integer,  (the number of workshops the student signed up for).
   * An array of type *Workshop* array having size .

Implement  [functions](http://www.cplusplus.com/doc/tutorial/functions/):

1. *Available\_Workshops\* initialize (int start\_time[], int duration[], int n)*  
   Creates an *Available\_Workshops* object and initializes its elements using the elements in the  and  parameters (both are of size ). Here,  and  are the respective start time and duration for the  workshop. This function must return a pointer to an *Available\_Workshops* object.
2. *int CalculateMaxWorkshops(Available\_Workshops\* ptr)*  
   Returns the maximum number of workshops the student can attend—without overlap. The next workshop cannot be attended until the previous workshop ends.

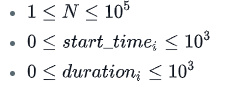
**Note:** An array of unknown size () should be declared as follows:

DataType\* arrayName = new DataType[n];

**Input Format**

Input from stdin is handled by the locked code in the editor; you simply need to write your functions to meet the specifications of the problem statement above.

**Constraints**



**Output Format**

Output to stdout is handled for you.

Your *initialize* function must return a pointer to an *Available\_Workshops* object.  
Your *CalculateMaxWorkshops* function must return maximum number of non-overlapping workshops the student can attend.

**Sample Input**

6

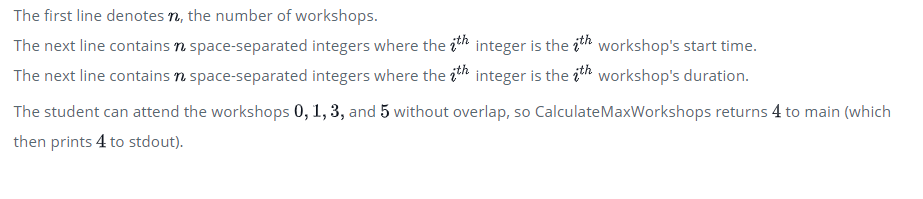
1 3 0 5 5 8

1 1 6 2 4 1

**Sample Output**

*CalculateMaxWorkshops* should return 4.

**Explanation**



<https://en.wikipedia.org/wiki/Interval_scheduling>