



ALVIN B 2024-CSE

A2

Started on Monday, 25 August 2025, 1:31 PM

State Finished

Completed on Monday, 25 August 2025, 2:02 PM

Time taken 30 mins 42 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Convert the following algorithm into a program and find its time complexity using the counter method.

```
void function (int n)
```

```
{
```

```
    int i = 1;
```

```
    int s = 1;
```

```
    while(s <= n)
```

```
{
```

```
    i++;
```

```
    s += i;
```

```
}
```

```
}
```

Note: No need of counter increment for declarations and `scanf()` and `count` variable `printf()` statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

For example:

Input	Result
9	12

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 void counter(int n) {
4     int c=1;
5     int i = 1;
6     int s = 1;
7
8
9
10 while (s <= n) {
11     i++;
12     s += i;
13     c+=3;
14 }
15 c+=2;
16 printf("%d\n", c);
17 }
18
19 int main() {
20     int n;
21     scanf("%d", &n);
22     counter(n);
23     return 0;
24 }
25
```

	Input	Expected	Got	
✓	9	12	12	✓
✓	4	9	9	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Monday, 25 August 2025, 1:51 PM

State Finished

Completed on Monday, 25 August 2025, 2:15 PM

Time taken 23 mins 34 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Convert the following algorithm into a program and find its time complexity using the counter method.

```
void func(int n)
{
    if(n==1)
    {
        printf("*");
    }
    else
    {
        for(int i=1; i<=n; i++)
        {
            for(int j=1; j<=n; j++)
            {
                printf("*");
                printf("*");
                break;
            }
        }
    }
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 void func(int n)
3 {
4     int count = 2;
5     if (n == 1) {
6         count++;
7     } else {
8         for (int i = 1; i <= n; i++) {
9             count++;
10        for (int j = 1; j <= n; j++) {
11            count+=4;
12            break;
13        }
14    }
15 }
16
17 printf("%d", count);
18
19 }
20
21 int main() {
22     int n;
23     scanf("%d", &n);
24     func(n);
25 }
26
27
```

	Input	Expected	Got	
✓	2	12	12	✓
✓	1000	5002	5002	✓
✓	143	717	717	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Monday, 25 August 2025, 2:15 PM**State** Finished**Completed on** Monday, 25 August 2025, 2:24 PM**Time taken** 8 mins 11 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Convert the following algorithm into a program and find its time complexity using counter method.

```
Factor(num) {
{
    for (i = 1; i <= num; ++i)
    {
        if (num % i == 0)
        {
            printf("%d ", i);
        }
    }
}
```

Note: No need of counter increment for declarations and scanf() and counter variable printf() statement.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```
1 #include <stdio.h>
2
3 void Factor(int num) {
4     int c=1;
5     for (int i = 1; i <= num; ++i) {
6         c+=2;
7         if (num % i == 0) {
8             c++;
9
10        }
11    }
12    printf("%d",c);
13 }
14
15 int main() {
16     int num;
17     scanf("%d", &num);
18     Factor(num);
19     return 0;
20 }
21
```

	Input	Expected	Got	
✓	12	31	31	✓
✓	25	54	54	✓
✓	4	12	12	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Monday, 25 August 2025, 2:24 PM

State Finished

Completed on Monday, 25 August 2025, 2:28 PM

Time taken 3 mins 41 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Convert the following algorithm into a program and find its time complexity using counter method.

```
void function(int n)
{
    int c = 0;
    for(int i=n/2; i<n; i++)
        for(int j=1; j<n; j = 2 * j)
            for(int k=1; k<n; k = k * 2)
                c++;
}
```

Note: No need of counter increment for declarations and `scanf()` and `count` variable `printf()` statements.

Input:

A positive Integer `n`

Output:

Print the value of the counter variable

Answer:

```
1 #include <stdio.h>
2
3 void function(int n) {
4     int c = 2;
5
6     for (int i = n / 2; i < n; i++) {
7         c+=2;
8         for (int j = 1; j < n; j = 2 * j) {
9             c+=2;
10            for (int k = 1; k < n; k = k * 2) {
11                c+=2;
12            }
13        }
14    }
15    printf("%d\n", c);
16 }
17
18 int main() {
19     int n;
20     scanf("%d", &n);
21     function(n);
22     return 0;
23 }
```

	Input	Expected	Got	
✓	4	30	30	✓
✓	10	212	212	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Monday, 25 August 2025, 2:28 PM**State** Finished**Completed on** Monday, 25 August 2025, 2:41 PM**Time taken** 12 mins 39 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Convert the following algorithm into a program and find its time complexity using counter method.

```
void reverse(int n)
{
    int rev = 0, remainder;
    while (n != 0)
    {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n /= 10;

    }
    print(rev);
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```
1 #include <stdio.h>
2
3 void reverse(int n) {
4     int rev = 0, remainder;
5     int c=1;
6     c++;
7     c++;
8     while (n != 0) {
9         remainder = n % 10;
10        rev = rev * 10 + remainder;
11        n /= 10;
12        c+=4;
13    }
14
15    printf("%d\n", c);
16 }
17
18 int main() {
19     int n;
20     scanf("%d", &n);
21     reverse(n);
22     return 0;
23 }
24
```

	Input	Expected	Got	
✓	12	11	11	✓
✓	1234	19	19	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Monday, 29 September 2025, 1:43 PM**State** Finished**Completed on** Monday, 29 September 2025, 2:02 PM**Time taken** 19 mins 13 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Problem Statement

Given an array of 1s and 0s this has all 1s first followed by all 0s. Aim is to find the number of 0s. Write a program using Divide and Conquer to Count the number of zeroes in the given array.

Input Format

First Line Contains Integer m – Size of array

Next m lines Contains m numbers – Elements of an array

Output Format

First Line Contains Integer – Number of zeroes present in the given array.

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 int findFirstZero(int arr[], int low, int high) {
4     if (high >= low) {
5         int mid = low + (high - low) / 2;
6
7         if ((mid == 0 || arr[mid - 1] == 1) && arr[mid] == 0) {
8             return mid;
9         }
10
11        if (arr[mid] == 1) {
12            return findFirstZero(arr, mid + 1, high);
13        }
14        else {
15            return findFirstZero(arr, low, mid - 1);
16        }
17    }
18    return -1;
19 }
20 int countZeroes(int arr[], int n) {
21     int firstZeroIndex = findFirstZero(arr, 0, n - 1);
22
23     if (firstZeroIndex == -1) {
24         return 0;
25     }
26     return (n - firstZeroIndex);
27 }
28
29 int main() {
30     int m;
31     scanf("%d", &m);
32
33     int arr[m];
34     for (int i = 0; i < m; i++) {
35         scanf("%d", &arr[i]);
36     }
37
38     printf("%d\n", countZeroes(arr, m));
39
40     return 0;
41 }
```

	Input	Expected	Got	
✓	5 1 1 1 0 0	2	2	✓

	Input	Expected	Got	
✓	10 1 1 1 1 1 1 1 1 1 1	0	0	✓
✓	8 0 0 0 0 0 0 0 0 0	8	8	✓
✓	17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0	2	2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Monday, 29 September 2025, 2:02 PM

State Finished

Completed on Monday, 29 September 2025, 2:09 PM

Time taken 6 mins 53 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than $\lfloor n / 2 \rfloor$ times. You may assume that the majority element always exists in the array.

Example 1:

Input: `nums = [3,2,3]`
Output: 3

Example 2:

Input: `nums = [2,2,1,1,1,2,2]`
Output: 2

Constraints:

- `n == nums.length`
- `1 <= n <= 5 * 10^4`
- `-2^31 <= nums[i] <= 2^31 - 1`

For example:

Input	Result
3	3
3 2 3	
7	2
2 2 1 1 1 2 2	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 int majorityElement(int nums[], int n) {
4     int candidate = 0;
5     int count = 0;
6     for (int i = 0; i < n; i++) {
7         if (count == 0) {
8             candidate = nums[i];
9             count = 1;
10        } else if (nums[i] == candidate) {
11            count++;
12        } else {
13            count--;
14        }
15    }
16
17    return candidate;
18}
19
20 int main() {
21     int n, i;
22
23     if (scanf("%d", &n) != 1 || n <= 0) {
24         printf("Invalid input for array size.\n");
25         return 1;
26     }
27     int* nums = (int*)malloc(n * sizeof(int));
28     if (nums == NULL) {
29         printf("Memory allocation failed.\n");
30         return 1;
31     }
32
33     for (i = 0; i < n; i++) {
34         scanf("%d", &nums[i]);
35     }
36 }
```

```
35 }  
36     int result = majorityElement(nums, n);  
37     printf("%d\\n", result);  
38     free(nums);  
39  
40     return 0;  
41 }
```

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Monday, 29 September 2025, 2:10 PM

State Finished

Completed on Monday, 29 September 2025, 2:13 PM

Time taken 2 mins 55 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00**Problem Statement:**

Given a sorted array and a value x, the floor of x is the largest element in array smaller than or equal to x. Write divide and conquer algorithm to find floor of x.

Input Format

First Line Contains Integer n – Size of array
 Next n lines Contains n numbers – Elements of an array
 Last Line Contains Integer x – Value for x

Output Format

First Line Contains Integer – Floor value for x

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int findFloor(int arr[], int n, int x) {
5     int low = 0;
6     int high = n - 1;
7     int floor_val = -1;
8     while (low <= high) {
9         int mid = low + (high - low) / 2;
10    if (arr[mid] == x) {
11        return arr[mid];
12    } else if (arr[mid] < x) {
13        floor_val = arr[mid];
14        low = mid + 1;
15    } else {
16        high = mid - 1;
17    }
18 }
19 return floor_val;
20 }
21
22 int main() {
23     int n, i, x;
24     scanf("%d", &n);
25     int* arr = (int*)malloc(n * sizeof(int));
26     for (i = 0; i < n; i++) {
27         scanf("%d", &arr[i]);
28     }
29     scanf("%d", &x);
30     int floor_value = findFloor(arr, n, x);
31     printf("%d\n", floor_value);
32     free(arr);
33     return 0;
34 }
35

```

	Input	Expected	Got	
✓	6	2	2	✓
	1			
	2			
	8			
	10			
	12			
	19			
	5			

	Input	Expected	Got	
✓	5 10 22 85 108 129 100	85	85	✓
✓	7 3 5 7 9 11 13 15 10	9	9	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Monday, 29 September 2025, 2:13 PM**State** Finished**Completed on** Monday, 29 September 2025, 2:22 PM**Time taken** 8 mins 29 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00**Problem Statement:**

Given a sorted array of integers say arr[] and a number x. Write a recursive program using divide and conquer strategy to check if there exist two elements in the array whose sum = x. If there exist such two elements then return the numbers, otherwise print as "No".

Note: Write a Divide and Conquer Solution

Input Format

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Sum Value

Output Format

First Line Contains Integer – Element1

Second Line Contains Integer – Element2 (Element 1 and Elements 2 together sums to value "x")

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int find_pair_recursive(int arr[], int low, int high, int x) {
5     if (low >= high) {
6         return 0;
7     }
8
9     int sum = arr[low] + arr[high];
10
11    if (sum == x) {
12        printf("%d\n", arr[low]);
13        printf("%d\n", arr[high]);
14        return 1;
15    } else if (sum < x) {
16        return find_pair_recursive(arr, low + 1, high, x);
17    } else {
18        return find_pair_recursive(arr, low, high - 1, x);
19    }
20}
21
22 int main() {
23     int n, i, x;
24     scanf("%d", &n);
25     int* arr = (int*)malloc(n * sizeof(int));
26     if (arr == NULL) {
27         return 1;
28     }
29     for (i = 0; i < n; i++) {
30         scanf("%d", &arr[i]);
31     }
32     scanf("%d", &x);
33
34     if (!find_pair_recursive(arr, 0, n - 1, x)) {
35         printf("No\n");
36     }
37
38     free(arr);
39     return 0;
40 }
41
```

	Input	Expected	Got	
✓	4 2 4 8 10 14	4 10	4 10	✓
✓	5 2 4 6 8 10 100	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



7 ALVIN B 2024-CSE

A2

Started on Monday, 29 September 2025, 2:22 PM**State** Finished**Completed on** Monday, 29 September 2025, 2:37 PM**Time taken** 15 mins 22 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Write a Program to Implement the Quick Sort Algorithm

Input Format:

The first line contains the no of elements in the list-n

The next n lines contain the elements.

Output:

Sorted list of elements

For example:

Input	Result
5	12 34 67 78 98
67 34 12 98 78	

Answer:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void swap(int* a, int* b) {
5     int temp = *a;
6     *a = *b;
7     *b = temp;
8 }
9
10 int partition(int arr[], int low, int high) {
11     int pivot = arr[high];
12     int i = (low - 1);
13     int j;
14
15     for (j = low; j < high; j++) {
16         if (arr[j] <= pivot) {
17             i++;
18             swap(&arr[i], &arr[j]);
19         }
20     }
21     swap(&arr[i + 1], &arr[high]);
22     return (i + 1);
23 }
24
25 void quickSort(int arr[], int low, int high) {
26     if (low < high) {
27         int pi = partition(arr, low, high);
28         quickSort(arr, low, pi - 1);
29         quickSort(arr, pi + 1, high);
30     }
31 }
32
33 void printArray(int arr[], int size) {
34     int i;
35     for (i = 0; i < size; i++) {
36         printf("%d ", arr[i]);
37     }
38     printf("\n");
39 }
40
41 int main() {
42     int n, i;
43     scanf("%d", &n);
44     int* arr = (int*)malloc(n * sizeof(int));
45     if (arr == NULL) {
46         return 1;
47     }
48     for (i = 0; i < n; i++) {
49         scanf("%d", &arr[i]);
50     }
51     quickSort(arr, 0, n - 1);

```

	Input	Expected	Got	
✓	5 67 34 12 98 78	12 34 67 78 98	12 34 67 78 98	✓
✓	10 1 56 78 90 32 56 11 10 90 114	1 10 11 32 56 56 78 90 90 114	1 10 11 32 56 56 78 90 90 114	✓
✓	12 9 8 7 6 5 4 3 2 1 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Monday, 25 August 2025, 2:41 PM**State** Finished**Completed on** Monday, 25 August 2025, 2:47 PM**Time taken** 6 mins 21 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Write a program to take value V and we want to make change for V Rs, and we have infinite supply of each of the denominations in Indian currency, i.e., we have infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or notes needed to make the change.

Input Format:

Take an integer from stdin.

Output Format:

print the integer which is change of the number.

Example Input :

64

Output:

4

Explanation:

We need a 50 Rs note and a 10 Rs note and two 2 rupee coins.

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 void make_change(int V) {
4     int denominations[] = {1000, 500, 100, 50, 20, 10, 5, 2, 1};
5     int num_coins = 0;
6
7     for (int i = 0; i < 9; i++) {
8
9         num_coins += V / denominations[i];
10        V = V % denominations[i];
11
12        if (V == 0) break;
13    }
14
15    printf("%d\n", num_coins);
16 }
17
18 int main() {
19     int V;
20     scanf("%d", &V);
21     make_change(V);
22     return 0;
23 }
24

```

	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Monday, 25 August 2025, 2:47 PM

State Finished

Completed on Saturday, 30 August 2025, 8:31 PM

Time taken 5 days 5 hours

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie.

Each child i has a greed factor $g[i]$, which is the minimum size of a cookie that the child will be content with; and each cookie j has a size $s[j]$. If $s[j] \geq g[i]$, we can assign the cookie j to the child i , and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.

Example 1:**Input:**

```
3
1 2 3
2
1 1
```

Output:

```
1
```

Explanation: You have 3 children and 2 cookies. The greed factors of 3 children are 1, 2, 3.

And even though you have 2 cookies, since their size is both 1, you could only make the child whose greed factor is 1 content.

You need to output 1.

Constraints:

```
1 <= g.length <= 3 * 10^4
0 <= s.length <= 3 * 10^4
1 <= g[i], s[j] <= 2^31 - 1
```

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int compare(const void *a, const void *b) {
4     return (*(int*)a - *(int*)b);
5 }
6
7 int findContentChildren(int* g, int gSize, int* s, int sSize) {
8     qsort(g, gSize, sizeof(int), compare);
9     qsort(s, sSize, sizeof(int), compare);
10    int child_idx = 0;
11    int cookie_idx = 0;
12    int content_children = 0;
13    while (child_idx < gSize && cookie_idx < sSize) {
14        if (s[cookie_idx] >= g[child_idx]) {
15            content_children++;
16            child_idx++;
17            cookie_idx++;
18        } else {
19            cookie_idx++;
20        }
21    }
22
23    return content_children;
24 }
25
26
27 int main() {
28     int gSize;
29     scanf("%d", &gSize);
30     int g[gSize];
31     for (int i = 0; i < gSize; i++) {
32         scanf("%d", &g[i]);
33     }
34
35     int sSize;
36
37     scanf("%d", &sSize);
```

```
39     int s[sSize];
40     for (int i = 0; i < sSize; i++) {
41         scanf("%d", &s[i]);
42     }
43
44     int result = findContentChildren(g, gSize, s, sSize);
45     printf("%d", result);
46
47     return 0;
48 }
```

	Input	Expected	Got	
✓	2	2	2	✓
	1 2			
	3			
	1 2 3			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Saturday, 6 September 2025, 7:39 PM

State Finished

Completed on Saturday, 6 September 2025, 9:56 PM

Time taken 2 hours 16 mins

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

A person needs to eat burgers. Each burger contains a count of calorie. After eating the burger, the person needs to run a distance to burn out his calories.

If he has eaten i burgers with c calories each, then he has to run at least $3^i * c$ kilometers to burn out the calories. For example, if he ate 3

burgers with the count of calorie in the order: [1, 3, 2], the kilometers he needs to run are $(3^0 * 1) + (3^1 * 3) + (3^2 * 2) = 1 + 9 + 18 = 28$.

But this is not the minimum, so need to try out other orders of consumption and choose the minimum value. Determine the minimum distance

he needs to run. Note: He can eat burger in any order and use an efficient sorting algorithm. Apply greedy approach to solve the problem.

Input Format

First Line contains the number of burgers

Second line contains calories of each burger which is n space-separate integers

Output Format

Print: Minimum number of kilometers needed to run to burn out the calories

Sample Input

```
3
5 10 7
```

Sample Output

```
76
```

For example:

Test	Input	Result
Test Case 1	3 1 3 2	18

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2 #include <math.h>
3 int main() {
4     int n;
5     scanf("%d", &n);
6     int arr[n];
7     for(int i=0;i<n;i++){
8         scanf("%d",&arr[i]);
9     }
10    if(n==0){
11        printf("%d",0);
12    }
13    else{
14        int c=0;
15        for(int i=0;i<n;i++){
16            for(int j=0;j<n;j++){
17                if(arr[i]>arr[j]){
18                    int t=arr[i];
19                    arr[i]=arr[j];
20                    arr[j]=t;
21                }
22            }
23        }
24        for(int i=0;i<n;i++){
25            c=c+(pow(n,i)*arr[i]);
26        }
27    }
28    printf("%d",c);
}
```

29 | }

	Test	Input	Expected	Got	
✓	Test Case 1	3 1 3 2	18	18	✓
✓	Test Case 2	4 7 4 9 6	389	389	✓
✓	Test Case 3	3 5 10 7	76	76	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Saturday, 6 September 2025, 7:46 PM**State** Finished**Completed on** Saturday, 6 September 2025, 7:50 PM**Time taken** 4 mins 6 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Given an array of N integer, we have to maximize the sum of $\text{arr}[i] * i$, where i is the index of the element ($i = 0, 1, 2, \dots, N$). Write an algorithm based on Greedy technique with a Complexity $O(n\log n)$.

Input Format:

First line specifies the number of elements-n

The next n lines contain the array elements.

Output Format:

Maximum Array Sum to be printed.

Sample Input:

5

2 5 3 4 0

Sample output:

40

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 int compare(const void *a, const void *b) {
4     return (*(int *)a - *(int *)b);
5 }
6
7 int main() {
8     int n;
9     scanf("%d", &n);
10
11    int arr[n];
12    for(int i = 0; i < n; i++) {
13        scanf("%d", &arr[i]);
14    }
15
16    qsort(arr, n, sizeof(int), compare);
17
18    long long maxSum = 0;
19    for(int i = 0; i < n; i++) {
20        maxSum += (long long)arr[i] * i;
21    }
22
23    printf("%lld\n", maxSum);
24    return 0;
25 }
26

```

	Input	Expected	Got	
✓	5	40	40	✓
	2			
	5			
	3			
	4			
	0			

	Input	Expected	Got	
✓	10 2 2 2 4 4 3 3 5 5 5	191	191	✓
✓	2 45 3	45	45	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Saturday, 6 September 2025, 7:50 PM**State** Finished**Completed on** Saturday, 6 September 2025, 7:58 PM**Time taken** 7 mins 53 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Given two arrays array_One[] and array_Two[] of same size N. We need to first rearrange the arrays such that the sum of the product of pairs(1 element from each) is minimum. That is SUM (A[i] * B[i]) for all i is minimum.

For example:

Input	Result
3	28
1	
2	
3	
4	
5	
6	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 int compare_asc(const void *a, const void *b) {
4     return (*(int *)a - *(int *)b);
5 }
6 int compare_desc(const void *a, const void *b) {
7     return (*(int *)b - *(int *)a);
8 }
9
10 int main() {
11     int n;
12     scanf("%d", &n);
13
14     int array_One[n], array_Two[n];
15
16     for(int i = 0; i < n; i++) {
17         scanf("%d", &array_One[i]);
18     }
19
20     for(int i = 0; i < n; i++) {
21         scanf("%d", &array_Two[i]);
22     }
23     qsort(array_One, n, sizeof(int), compare_asc);
24     qsort(array_Two, n, sizeof(int), compare_desc);
25
26     long long min_sum = 0;
27     for(int i = 0; i < n; i++) {
28         min_sum += (long long)array_One[i] * array_Two[i];
29     }
30
31     printf("%lld\n", min_sum);
32     return 0;
33 }
34

```

	Input	Expected	Got	
✓	3	28	28	✓
	1			
	2			
	3			
	4			
	5			
	6			

	Input	Expected	Got	
✓	4 7 5 1 2 1 3 4 1	22	22	✓
✓	5 20 10 30 10 40 8 9 4 3 10	590	590	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Saturday, 1 November 2025, 12:08 AM

State Finished

Completed on Saturday, 1 November 2025, 12:28 AM

Time taken 19 mins 53 secs

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 10.00 out of 10.00

Playing with Numbers:

Ram and Sita are playing with numbers by giving puzzles to each other. Now it was Ram's turn, so he gave Sita a positive integer 'n' and two numbers 1 and 3. He asked her to find the possible ways by which the number n can be represented using 1 and 3. Write any efficient algorithm to find the possible ways.

Example 1:

Input: 6

Output: 6

Explanation: There are 6 ways to represent the number with 1 and 3

```
1+1+1+1+1+1
3+3
1+1+1+3
1+1+3+1
1+3+1+1
3+1+1+1
```

Input Format

First Line contains the number n

Output Format

Print: The number of possible ways 'n' can be represented using 1 and 3

Sample Input

6

Sample Output

6

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 static char *my_strdup(const char *s) {
6     size_t len = strlen(s);
7     char *r = malloc(len + 1);
8     if (!r) return NULL;
9     memcpy(r, s, len + 1);
10    return r;
11 }
12
13 static char *add_strings(const char *a, const char *b) {
14     int la = (int)strlen(a);
15     int lb = (int)strlen(b);
16     int i = la - 1, j = lb - 1;
17     int carry = 0;
18     int pos = 0;
19     int maxlen = (la > lb ? la : lb) + 1;
20     char *tmp = malloc(maxlen + 1);
21     if (!tmp) return NULL;
22
23     while (i >= 0 || j >= 0 || carry) {
24         int da = (i >= 0) ? (a[i] - '0') : 0;
25         int db = (j >= 0) ? (b[j] - '0') : 0;
26         int s = da + db + carry;
27         tmp[pos++] = (char)('0' + (s % 10));
28         carry = s / 10;
29         i--;
30     }
31     tmp[pos] = '\0';
32     for (int left = 0, right = pos - 1; left < right; left++, right--) {
33         char c = tmp[left];
34         tmp[left] = tmp[right];
35         tmp[right] = c;
36     }
37     char *res = malloc(pos + 1);
38     if (!res) { free(tmp); return NULL; }
```

```
39     memcpy(res, tmp, pos + 1);
40     free(tmp);
41     return res;
42 }
43
44 int main(void) {
45     int n;
46     if (scanf("%d", &n) != 1) return 0;
47     if (n < 0) { printf("0\n"); return 0; }
48     char **dp = calloc((size_t)n + 1, sizeof(char *));
49     if (!dp) return 0;
50
51     dp[0] = my_strdup("1");
52     for (int i = 1; i <= n; ++i) {
```

	Input	Expected	Got	
✓	6	6	6	✓
✓	25	8641	8641	✓
✓	100	24382819596721629	24382819596721629	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Tuesday, 28 October 2025, 8:22 AM

State Finished

Completed on Tuesday, 28 October 2025, 8:26 AM

Time taken 4 mins 30 secs

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 10.00 out of 10.00

Playing with Chessboard:

Ram is given with an $n \times n$ chessboard with each cell with a monetary value. Ram stands at the $(0,0)$, that the position of the top left white rook. He is been given a task to reach the bottom right black rook position $(n-1, n-1)$ constrained that he needs to reach the position by traveling the maximum monetary path under the condition that he can only travel one step right or one step down the board. Help ram to achieve it by providing an efficient DP algorithm.

Example:

Input

```
3
1 2 4
2 3 4
8 7 1
```

Output:

```
19
```

Explanation:

Totally there will be 6 paths among that the optimal is

Optimal path value: $1+2+8+7+1=19$

Input Format

First Line contains the integer n

The next n lines contain the $n \times n$ chessboard values

Output Format

Print Maximum monetary value of the path

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 #define MAX 100
4
5 int max(int a, int b) {
6     return (a > b) ? a : b;
7 }
8
9 int main() {
10    int n;
11    int grid[MAX][MAX];
12    int dp[MAX][MAX];
13    scanf("%d", &n);
14    for (int i = 0; i < n; i++) {
15        for (int j = 0; j < n; j++) {
16            scanf("%d", &grid[i][j]);
17        }
18    }
19
20    dp[0][0] = grid[0][0];
21
22    for (int j = 1; j < n; j++) {
23        dp[0][j] = dp[0][j - 1] + grid[0][j];
24    }
25
26    for (int i = 1; i < n; i++) {
27        dp[i][0] = dp[i - 1][0] + grid[i][0];
28    }
29
30    for (int i = 1; i < n; i++) {
31        for (int j = 1; j < n; j++) {
32            dp[i][j] = grid[i][j] + max(dp[i - 1][j], dp[i][j - 1]);
33        }
34    }
35
36    printf("%d\n", dp[n - 1][n - 1]);
```

```
37  
38     return 0;  
39 }  
40
```

	Input	Expected	Got	
✓	3 1 2 4 2 3 4 8 7 1	19	19	✓
✓	3 1 3 1 1 5 1 4 2 1	12	12	✓
✓	4 1 1 3 4 1 5 7 8 2 3 4 6 1 6 9 0	28	28	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)





ALVIN B 2024-CSE ▾

A2

Started on Tuesday, 28 October 2025, 8:27 AM

State Finished

Completed on Tuesday, 28 October 2025, 8:30 AM

Time taken 3 mins 15 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Given two strings find the length of the common longest subsequence(need not be contiguous) between the two.

Example:

s1: ggtabe

s2: tgatasb

s1	a	g	g	t	a	b	
s2	g	x	t	x	a	y	b

The length is 4

Solveing it using Dynamic Programming

For example:

Input	Result
aab	2
azb	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2 #include <string.h>
3
4 #define MAX 1000
5
6 int max(int a, int b) {
7     return (a > b) ? a : b;
8 }
9
10 int main() {
11     char s1[MAX], s2[MAX];
12     int dp[MAX+1][MAX+1];
13
14     scanf("%s", s1);
15     scanf("%s", s2);
16
17     int m = strlen(s1);
18     int n = strlen(s2);
19     for (int i = 0; i <= m; i++) {
20         for (int j = 0; j <= n; j++) {
21             if (i == 0 || j == 0)
22                 dp[i][j] = 0;
23             else if (s1[i - 1] == s2[j - 1])
24                 dp[i][j] = 1 + dp[i - 1][j - 1];
25             else
26                 dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
27         }
28     }
29     printf("%d\n", dp[m][n]);
30
31     return 0;
32 }
33

```

	Input	Expected	Got	
✓	aab azb	2	2	✓
✓	ABCD ABCD	4	4	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Tuesday, 28 October 2025, 8:30 AM

State Finished

Completed on Tuesday, 28 October 2025, 8:34 AM

Time taken 3 mins 30 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Problem statement:

Find the length of the Longest Non-decreasing Subsequence in a given Sequence.

Eg:

Input:9

Sequence:[-1,3,4,5,2,2,2,2,3]

the subsequence is [-1,2,2,2,2,3]

Output:6

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 #define MAX 1000
4
5 int max(int a, int b) {
6     return (a > b) ? a : b;
7 }
8
9 int main() {
10    int n;
11    int arr[MAX], dp[MAX];
12
13    scanf("%d", &n);
14    for (int i = 0; i < n; i++) {
15        scanf("%d", &arr[i]);
16    }
17
18    for (int i = 0; i < n; i++) {
19        dp[i] = 1;
20    }
21
22    for (int i = 1; i < n; i++) {
23        for (int j = 0; j < i; j++) {
24            if (arr[j] <= arr[i]) {
25                dp[i] = max(dp[i], dp[j] + 1);
26            }
27        }
28    }
29
30    int ans = dp[0];
31    for (int i = 1; i < n; i++) {
32        ans = max(ans, dp[i]);
33    }
34
35    printf("%d\n", ans);
36
37    return 0;
38 }
39

```

	Input	Expected	Got	
✓	9 -1 3 4 5 2 2 2 2 3	6	6	✓
✓	7 1 2 2 4 5 7 6	6	6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Tuesday, 28 October 2025, 8:35 AM**State** Finished**Completed on** Tuesday, 28 October 2025, 8:38 AM**Time taken** 3 mins 30 secs**Marks** 1.00/1.00**Grade** 4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5	1
1 1 2 3 4	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 #define MAX 100000
4
5 int main() {
6     int n;
7     int arr[MAX], count[MAX] = {0};
8     scanf("%d", &n);
9     for (int i = 0; i < n; i++) {
10         scanf("%d", &arr[i]);
11     }
12
13     for (int i = 0; i < n; i++) {
14         if (count[arr[i]] == 1) {
15             printf("%d\n", arr[i]);
16             return 0;
17         }
18         count[arr[i]] = 1;
19     }
20     printf("No duplicate found\n");
21     return 0;
22 }
```

	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Tuesday, 28 October 2025, 8:39 AM**State** Finished**Completed on** Tuesday, 28 October 2025, 8:42 AM**Time taken** 3 mins 37 secs**Marks** 1.00/1.00**Grade** 4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5	1
1 1 2 3 4	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 #define MAX 100000
4
5 int main() {
6     int n;
7     int arr[MAX];
8     int count[MAX] = {0};
9     scanf("%d", &n);
10    for (int i = 0; i < n; i++) {
11        scanf("%d", &arr[i]);
12    }
13    for (int i = 0; i < n; i++) {
14        if (count[arr[i]] == 1) {
15            printf("%d\n", arr[i]);
16            return 0;
17        }
18        count[arr[i]] = 1;
19    }
20    printf("No duplicate found\n");
21    return 0;
22 }
```

	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Tuesday, 28 October 2025, 8:42 AM**State** Finished**Completed on** Tuesday, 28 October 2025, 8:50 AM**Time taken** 7 mins 55 secs**Marks** 1.00/1.00**Grade** 30.00 out of 30.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

- The first line contains T, the number of test cases. Following T lines contain:

- Line 1 contains N1, followed by N1 integers of the first array
- Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1	10 57
3 10 17 57	
6	
2 7 10 15 57 246	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 int main() {
4     int T;
5     scanf("%d", &T);
6
7     while (T--) {
8         int n1, n2;
9         scanf("%d", &n1);
10        int a[n1];
11        for (int i = 0; i < n1; i++) {
12            scanf("%d", &a[i]);
13        }
14
15        scanf("%d", &n2);
16        int b[n2];
17        for (int i = 0; i < n2; i++) {
18            scanf("%d", &b[i]);
19        }
20
21        int i = 0, j = 0;
22        int printed = 0;

```

```

23
24    while (i < n1 && j < n2) {
25        if (a[i] == b[j]) {
26            if (printed) printf(" ");
27            printf("%d", a[i]);
28            printed = 1;
29            i++;
30            j++;
31        } else if (a[i] < b[j]) {
32            i++;
33        } else {
34            j++;
35        }
36    }
37    printf("\n");
38}
39
40    return 0;
41}
42
43

```

	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

Started on Tuesday, 28 October 2025, 8:51 AM**State** Finished**Completed on** Tuesday, 28 October 2025, 9:01 AM**Time taken** 9 mins 53 secs**Marks** 1.00/1.00**Grade** 30.00 out of 30.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

- The first line contains T, the number of test cases. Following T lines contain:

- Line 1 contains N1, followed by N1 integers of the first array
- Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1	10 57
3 10 17 57	
6	
2 7 10 15 57 246	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 int main() {
4     int T;
5     scanf("%d", &T);
6
7     while (T--) {
8         int n1, n2;
9         scanf("%d", &n1);
10        int a[n1];
11        for (int i = 0; i < n1; i++) {
12            scanf("%d", &a[i]);
13        }
14
15        scanf("%d", &n2);
16        int b[n2];
17        for (int i = 0; i < n2; i++) {
18            scanf("%d", &b[i]);
19        }
20
21        int i = 0, j = 0;
22        int printed = 0;

```

```

23
24    while (i < n1 && j < n2) {
25        if (a[i] == b[j]) {
26            if (!printed) {
27                printf("%d", a[i]);
28                printed = 1;
29            } else {
30                printf(" %d", a[i]);
31            }
32            i++;
33            j++;
34        } else if (a[i] < b[j]) {
35            i++;
36        } else {
37            j++;
38        }
39    }
40    printf("\n");
41}
42
43    return 0;
44}
45

```

	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Tuesday, 28 October 2025, 9:01 AM**State** Finished**Completed on** Tuesday, 28 October 2025, 9:10 AM**Time taken** 9 mins 14 secs**Marks** 1.00/1.00**Grade** 4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that $A[j] - A[i] = k$, $i \neq j$.

Input Format:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as $5 - 1 = 4$

So Return 1.

For example:

Input	Result
3	1
1 3 5	
4	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int A[n];
8     for (int i = 0; i < n; i++) {
9         scanf("%d", &A[i]);
10    }
11
12    int k;
13    scanf("%d", &k);
14
15    int i = 0, j = 1;
16    int found = 0;
17
18    while (j < n) {
19        int diff = A[j] - A[i];
20        if (diff == k && i != j) {
21            found = 1;
22            break;
23        } else if (diff < k) {
24            j++;
25        } else {
26            i++;
27            if (i == j) j++;
28        }
29    }
30
31    printf("%d\n", found);
32    return 0;
33}
34

```

	Input	Expected	Got	
✓	3 1 3 5 4	1	1	✓
✓	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✓
✓	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✓
✓	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE

A2

Started on Tuesday, 28 October 2025, 9:10 AM

State Finished

Completed on Tuesday, 28 October 2025, 9:18 AM

Time taken 7 mins 48 secs

Marks 1.00/1.00

Grade 4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that $A[j] - A[i] = k$, $i \neq j$.

Input Format:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as $5 - 1 = 4$

So Return 1.

For example:

Input	Result
3	1
1 3 5	
4	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int A[n];
8     for (int i = 0; i < n; i++) {
9         scanf("%d", &A[i]);
10    }
11
12    int k;
13    scanf("%d", &k);
14
15    int i = 0, j = 1;
16    int found = 0;
17
18    while (i < n && j < n) {
19        int diff = A[j] - A[i];
20
21        if (diff == k && i != j) {
22            found = 1;
23            break;
24        } else if (diff < k) {
25            j++;
26        } else {
27            i++;
28            if (i == j) j++;
29        }
30    }
31
32    printf("%d\n", found);
33    return 0;
34}
35

```

	Input	Expected	Got	
✓	3 1 3 5 4	1	1	✓
✓	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✓
✓	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✓
✓	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)