RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM 602 105

CS23331 Design and Analysis of Algorithms

Laboratory Record Note Book

Name : **LOHESH M**

Year / Branch / Section : **2025-26/Computer Science And Engineering**

University Register No. : **2116240701292**

College Roll No. : **240701292** .

Semester : **IIIrd Semester** .

Academic Year : **2025-26** .

RAJALAKSHMI ENGINEERING COLLEGE

An Autonomous Institution

BONAFIDE CERTIFICATE

Name: ……**LOHESH M**……

Academic Year: .**2025-26**. Semester: ……**III**…… Branch: **Computer Science And Engineering**

Register No. **2116240701292**

Certified that this is the bonafide record of work done by the above student in the.....**Design And Analysis Of Algorithms**.....Laboratory during the academic year 2025- 2026

Signature of Faculty in-charge

Submitted for the Practical Examination held on…**21st November, 2025**

Internal Examiner External Examiner

INDEX

EX.NO	DATE	NAME OF THE EXPERIMENT	GITHUB QR
1	30.07.2025	Basic C Programming	
2	17.08.2025	Time Complexity	
3	30.08.2025	Divide and conquer	
4	13.09.2025	Greedy Technique	
5	08.10.2025	Dynamic Programming	
6.	22.10.2025	Competitive Programming	

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 6 A
State	Finished
Completed on	Wednesday, 6 August 2025, 8:54 AM
Time taken	34 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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 %)

```
#include<stdio.h>

void function(int n,int* count){
    int i=1;
    ++(*count);
    int s=1;
    ++(*count);
    (*count)++;
    while(s<=n){
        ++(*count);
        i++;
        ++(*count);
        s+=i;
        ++(*count);
        // printf("%d\n",count);
    }
}

int main(){
    int count=0;
    int n;
    scanf("%d",&n);
    function(n,&count);
    printf("%d\n",count);
    return 0;
}
```

	Input	Expected	Got	
	9	12	12	
	4	9	9	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Back to Course

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 6 A
State	Finished
Completed on	Wednesday, 6 August 2025, 8:47 AM
Time taken	17 mins 30 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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 %)

```
#include<stdio.h>

void function(int n,int *count);

int main(){
    int count=0;
    int n;
    scanf("%d",&n);
    function(n,&count);
    printf("%d\n",count);
    return 0;
}

void function(int n,int *count){
    (*count)++;
    if(n==1){
        // printf("*");
        (*count)++;
    }
    else{
        (*count)++;
        for(int i=1;i<=n;i++){
            (*count)++;
            for(int j=1;j<=n;j++){
                (*count)++;
                // printf("*");
                (*count)++;
                // printf("*");
```

```
                (*count)++;
                (*count)++;
                break;

            }
        }
    }
}


```

	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 13
State	Finished
Completed on	Wednesday, 13 August 2025, 8:58 AM
Time taken	31 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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:

```c
#include<stdio.h>

void factor(int* num,long int* c){
    (*c)++;
    for(int i=1;i<=*num;i++){
        (*c)++;
        (*c)++;
        if(*num%i==0) { /*printf("%d ",i);*/ (*c)++;}
    }
    // printf("\n");
}

int main(){
    long int c=0;
    int n;
    scanf("%d",&n);
    factor(&n,&c);
    printf("%ld\n",c);
    return 0;
}
```

	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

undefined ×

Done

Opened: Wednesday, 24 September 2025, 10:08 AM

Attempts allowed: 2

Grading method: Highest grade

Summary of your previous attempts

Attempt	State	Marks / 1.00	Grade / 10.00	Review
1	Finished Submitted Wednesday, 13 August 2025, 9:12 AM	1.00	10.00	Review

Highest grade: 10.00 / 10.00.

Back to Course

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 13
State	Finished
Completed on	Wednesday, 13 August 2025, 9:16 AM
Time taken	3 mins 48 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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:

```c
#include<stdio.h>

void reverse(int n,long int* c)
{
   int rev = 0, remainder;
   (*c)+=2;
   while (n != 0)
    {
        (*c)++;
        remainder = n % 10;
        (*c)++;
        rev = rev * 10 + remainder;
        (*c)++;
        n/= 10;
        (*c)++;

    }
    // printf("%d",rev);
    (*c)++;
}

int main(){
    long int c=0;
    int n;
    scanf("%d",&n);
    reverse(n,&c);
    printf("%ld\n",c);
    return 0;
}
```

	Input	Expected	Got	
	12	11	11	
	1234	19	19	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Back to Course

undefined ×

Done

Opened: Tuesday, 2 September 2025, 11:34 AM

Grading method: Highest grade

Summary of your previous attempts

Attempt	State	Marks / 1.00	Grade / 10.00	Review
1	Finished Submitted Wednesday, 17 September 2025, 9:02 AM	1.00	10.00	Review

Highest grade: 10.00 / 10.00.

Back to Course

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 17 undefined × AM
State	Finished
Completed on	Wednesday, 17 September 2025, 9:58 PM
Time taken	12 hours 56 mins
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 `⌊n / 2⌋` 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 2 3	3
7 2 2 1 1 1 2 2	2

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdlib.h>

int boyer_moore_majority_voting(int* a, int n) {
    if (n == 0) return -1;

    int votes = 0;
    int candidate = -1;

    // First pass to find the candidate
    for (int i = 0; i < n; i++) {
        if (votes == 0) {
            candidate = a[i];
            votes = 1;
        } else {
            if (a[i] == candidate) votes++;
            else votes--;
        }
    }

    // Second pass to verify the candidate
    votes = 0;
    for (int i = 0; i < n; i++) {
        if (a[i] == candidate) votes++;
    }

```

```c
    // If votes for candidate are more than half, it's the majority
    if (votes > n / 2) {
        return candidate;
    } else {
        return -1; // No majority element
    }
}

int main() {
    int* a = NULL;
    int input;
    scanf("%d",&input);
    int count = 0;
    int capacity = 0;

    // Input reading and dynamic array allocation
    while (scanf("%d", &input) == 1) {
        if (count >= capacity) {
            capacity = (capacity == 0) ? 1 : capacity * 2;
            int* temp = (int*)realloc(a, capacity * sizeof(int));

            if (temp == NULL) {
                printf("Memory reallocation failed! Exiting.\n");
                free(a);
                return 1;
            }
```

	Input	Expected	Got	
	3 3 2 3	3	3	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Back to Course

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 17 PM
State	Finished
Completed on	Tuesday, 23 September 2025, 6:44 PM
Time taken	5 days 20 hours
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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 %)

```c
#include <stdio.h>
#include <stdlib.h>

int find_floor_rec(int *arr, int left, int right, int x) {
    if (left > right) return -1;

    int mid = left + (right - left) / 2;
    int val = *(arr + mid);

    if (val == x) {
        return mid;
    } else if (val < x) {
        /* Candidate mid might be floor; search right half for a clo
        int rightAns = find_floor_rec(arr, mid + 1, right, x);
        return (rightAns != -1) ? rightAns : mid;
    } else {
        /* val > x: floor must be in left half */
        return find_floor_rec(arr, left, mid - 1, x);
    }
}

int main(void) {
    int n, x;
    // printf("Enter number of elements: ");
    if (scanf("%d", &n) != 1 || n <= 0) {
        fprintf(stderr, "Invalid size\n");
        return 1;
    }

    int *arr = (int *)malloc(n * sizeof(int));
    if (!arr) {
        fprintf(stderr, "Memory allocation failed\n");
        return 1;
    }

    // printf("Enter %d sorted integers in ascending order:\n", n);
    for (int i = 0; i < n; ++i) {
        if (scanf("%d", arr + i) != 1) {
            fprintf(stderr, "Invalid input\n");
            free(arr);
            return 1;
        }
    }

    // printf("Enter value x: ");
    if (scanf("%d", &x) != 1) {
        fprintf(stderr, "Invalid input\n");
        free(arr);
        return 1;
    }
```

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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Tuesday, 23 Se
State	Finished
Completed on	Tuesday, 23 September 2025, 6:45 PM
Time taken	1 min 20 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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 %)

```c
#include <stdio.h>
#include <stdlib.h>

/* Recursive binary search on arr between indices left and right (in
   Returns index of target if found, -1 otherwise. */
int binary_search_rec(int *arr, int left, int right, int target) {
    if (left > right) return -1;
    int mid = left + (right - left) / 2;
    int val = *(arr + mid);
    if (val == target) return mid;
    if (val < target) return binary_search_rec(arr, mid + 1, right,
    return binary_search_rec(arr, left, mid - 1, target);
}

/* Divide and conquer to find two elements whose sum equals x.
   Searches in arr between indices left and right (inclusive).
   If found, stores the pair in *a and *b and returns 1, otherwise r
int find_pair_rec(int *arr, int left, int right, int x, int *a, int
    if (left >= right) return 0; /* at least two elements needed */

    int mid = left + (right - left) / 2;

    /* 1) Recurse on left half */
    if (find_pair_rec(arr, left, mid, x, a, b)) return 1;

    /* 2) Recurse on right half */
    if (find_pair_rec(arr, mid + 1, right, x, a, b)) return 1;

    /* 3) Check cross pairs: for each element in left half,
          look for (x - arr[i]) in the right half using recursive bi
    for (int i = left; i <= mid; ++i) {
        int need = *(arr + i);
        int complement = x - need;
        int j = binary_search_rec(arr, mid + 1, right, complement);
        if (j != -1) {
            *a = need;
            *b = *(arr + j);
            return 1;
        }
    }

    return 0;
}

int main(void) {
    int n;
    if (scanf("%d", &n) != 1 || n <= 0) {
        fprintf(stderr, "Invalid size\n");
        return 1;
    }

```

52

	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Tuesday, 23 Se
State	Finished
Completed on	Tuesday, 23 September 2025, 6:52 PM
Time taken	6 mins 13 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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 67 34 12 98 78	12 34 67 78 98

Answer:

```c
#include <stdio.h>
#include <stdlib.h>

int compare(const void *pa, const void *pb) {
    int a = *(const int *)pa;
    int b = *(const int *)pb;
    if (a < b) return -1;
    if (a > b) return 1;
    return 0;
}

int main(void) {
    int n;
    if (scanf("%d", &n) != 1 || n <= 0) return 1;

    int *a = malloc(sizeof(int) * n);
    if (!a) return 1;

    for (int i = 0; i < n; ++i) {
        if (scanf("%d", a + i) != 1) {
            free(a);
            return 1;
        }
    }

    qsort(a, n, sizeof(*a), compare);

    for (int i = 0; i < n; ++i) printf("%d ", a[i]);
    printf("\n");

    free(a);
    return 0;
}

```

	Input	Expected	Got	
	5 67 34 12 98 78	12 34 67 78 98	12 34 67 78 98	

	Input	Expected	Got	
	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 13
State	Finished
Completed on	Wednesday, 13 August 2025, 9:36 AM
Time taken	20 mins 22 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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

Explanaton:

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

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n, count = 0;
    scanf("%d", &n);

    int arr[] = {1, 2, 5, 10, 20, 50, 100, 500, 1000};
    int size = sizeof(arr)/sizeof(arr[0]);

    for (int i = size - 1; i >= 0 && n > 0; i--) {
        count += n / arr[i];
        n = n % arr[i];
    }

    printf("%d\n", count);
    return 0;
}
```

	Input	Expected	Got	
	49	5	5	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Back to Course

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Monday, 1 Sept
State	Finished
Completed on	Monday, 1 September 2025, 10:37 AM
Time taken	26 mins 48 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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] >= 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 %)

```c
#include<stdio.h>
#include<stdlib.h>

int compare(const void* a,const void* b){
    return (*(int*)a - *(int*)b);
}

int main(){
    int n; //no of children
    scanf("%d",&n);
    int child[n];
    for(int i=0;i<n;i++){
        scanf("%d",&child[i]);
    }

    int m;//no of cookies
    scanf("%d",&m);
    int cookies[m];
    for(int i=0;i<m;i++) scanf("%d",&cookies[i]);

    qsort(child,n,sizeof(child[0]),compare);
    qsort(cookies,m,sizeof(cookies[0]),compare);

    int content = 0;
    int child_idx = 0;
    int cookie_idx = 0;

    while (child_idx < n && cookie_idx < m) {
        if (cookies[cookie_idx] >= child[child_idx]) {
            content++;
            child_idx++;
            cookie_idx++;
```

```
        }
        else {
            cookie_idx++;
        }
    }

    printf("%d",content);

    return 0;
}
```

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

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Back to Course

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Monday, 1 Sept
State	Finished
Completed on	Monday, 1 September 2025, 11:02 AM
Time taken	24 mins 42 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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 %)

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

int compare(const void* a,const void* b){
    int a1= *(const int*)a, b1=*(const int*)b;
    return b1-a1;
}

int main(){
    int n;//no of burgers
    scanf("%d",&n);
    int calories[n];//calories of each burger
    for(int i=0;i<n;i++) scanf("%d",&calories[i]);

    qsort(calories,n,sizeof(calories[0]),compare);

    int kilometers=0;

    for(int i=0;i<n;i++) kilometers+=pow(n,i)*calories[i];
```

```
    printf("%d\n",kilometers);
    return 0;
}
```

	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

undefined ×

Started on	Monday, 1 Sept
State	Finished
Completed on	Monday, 1 September 2025, 11:28 AM
Time taken	7 mins 27 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 arr[i] * i, where i is the index of the element (i = 0, 1, 2, ..., N).Write an algorithm based on Greedy technique with a Complexity O(nlogn).

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 %)

```c
#include<stdio.h>
#include<stdlib.h>

int compare(const void* a,const void* b){
    return (*(int*)a-*(int*)b);
}

int main(){
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++) scanf("%d",&arr[i]);

    qsort(arr,n,sizeof(arr[0]),compare);

    int sum=0;
    for(int i=0;i<n;i++) sum+=arr[i]*i;

    printf("%d\n",sum);
    return 0;
}
```

	Input	Expected	Got	
	5 2 5 3 4 0	40	40	

	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Monday, 1 Sept
State	Finished
Completed on	Monday, 1 September 2025, 11:35 AM
Time taken	6 mins 3 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

undefined ×

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 1 2 3 4 5 6	28

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdlib.h>

// Comparison function for ascending order
int compare1(const void* a, const void* b) {
    return (*(int*)a - *(int*)b);
}

// Comparison function for descending order
int compare2(const void* a, const void* b) {
    return (*(int*)b - *(int*)a);
}

int main() {
    int n;
    scanf("%d", &n);

    // Dynamic allocation for arrays to handle variable size correct
    int* arr = (int*)malloc(n * sizeof(int));
    int* brr = (int*)malloc(n * sizeof(int));

    if (arr == NULL || brr == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    for (int i = 0; i < n; i++) {
        scanf("%d", &brr[i]);
    }

    // Sort the first array in ascending order
    qsort(arr, n, sizeof(int), compare1);
    // Sort the second array in descending order
    qsort(brr, n, sizeof(int), compare2);

    long long sum = 0; // Use long long to avoid potential integer o
    for (int i = 0; i < n; i++) {
        sum += (long long)arr[i] * brr[i];
    }

    printf("%lld\n", sum);

    free(arr);
    free(brr);
```

```
    return 0;
}
```

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

RAJALAKSHMI ENGINEERING COLLEGE

4 LOHESH M 2024-CSE L2

Started on	Wednesday, 8 October 2025, 8:25 AM
State	Finished
Completed on	Wednesday, 8 October 2025, 9:10 AM
Time taken	45 mins
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 term, 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 6 represent 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 %)

```
#include<stdio.h>
#include<stdlib.h>

long int memo[1000] = {0};

long int countWays(int n) {
    if (n == 0) return 1;
    if (n < 0) return 0;
    if (memo[n] != 0) return memo[n];  // already computed

    memo[n] = countWays(n - 1) + countWays(n - 3);
    return memo[n];
}

long int countWaysIter(int n){
    long int W[n+1];
    W[0] = 1;
    for (int i = 1; i <= n; i++) {
        W[i] = 0;
        if (i - 1 >= 0) W[i] += W[i - 1];
        if (i - 3 >= 0) W[i] += W[i - 3];
    }
    return W[n];
```

```
}

int main(){
    int n;
    scanf("%d",&n);
    printf("%ld\n",countWaysIter(n));
    return 0;
}
```


	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

RAJALAKSHMI ENGINEERING COLLEGE

4 LOHESH M 2024-CSE L2

Started on	Wednesday, 8 October 2025, 9:14 AM
State	Finished
Completed on	Wednesday, 8 October 2025, 9:23 PM
Time taken	12 hours 8 mins
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*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*n chessboard values

Output Format

Print Maximum monetary value of the path

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
#include <stdlib.h>

int n;

long int optimalPath(int (*board)[n]) {
    long int (*dp)[n] = calloc(n, sizeof *dp);

    dp[0][0] = board[0][0];

    int i, j;

    // Fill first row
    j = 1;
    while (j < n) {
        dp[0][j] = board[0][j] + dp[0][j - 1];
        j++;
    }

    // Fill first column
    i = 1;
    while (i < n) {
        dp[i][0] = board[i][0] + dp[i - 1][0];
        i++;
    }

    // Fill the rest of the table
    i = 1;
    while (i < n) {
        j = 1;
        while (j < n) {
```

```
            if (dp[i - 1][j] > dp[i][j - 1])
                dp[i][j] = board[i][j] + dp[i - 1][j];
            else
                dp[i][j] = board[i][j] + dp[i][j - 1];
            j++;
        }
        i++;
    }

    long int result = dp[n - 1][n - 1];
    free(dp);
```



```
    free(dp);
    return result;
}

int main() {
    scanf("%d", &n);

    int (*board)[n] = calloc(n, sizeof *board);

    int i = 0, j = 0;
    while (i < n) {
```

	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

RAJALAKSHMI ENGINEERING COLLEGE

4 LOHESH M 2024-CSE L2

Started on	Wednesday, 8 October 2025, 10:05 PM
State	Finished
Completed on	Wednesday, 8 October 2025, 10:14 PM
Time taken	8 mins 33 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 azb	2

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int longestCommonSubsequence(char *s1, char *s2) {
    int n = strlen(s1);
    int m = strlen(s2);

    // Allocate DP table dynamically
    int (*dp)[m + 1] = calloc(n + 1, sizeof *dp);

    int i = 1, j;
    while (i <= n) {
        j = 1;
        while (j <= m) {
            if (s1[i - 1] == s2[j - 1])
                dp[i][j] = dp[i - 1][j - 1] + 1;
            else
                dp[i][j] = dp[i - 1][j] > dp[i][j - 1] ? dp[i - 1][j] : dp[i][j - 1];
            j++;
        }
        i++;
    }

    int result = dp[n][m];
    free(dp);
    return result;
}

int main() {
    char s1[1000], s2[1000];

    scanf("%s", s1);
```

```
    scanf("%s", s2);

    printf("%d\n", longestCommonSubsequence(s1, s2));
    return 0;
}

```


	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

RAJALAKSHMI ENGINEERING COLLEGE

4 LOHESH M 2024-CSE L2

Started on	Wednesday, 8 October 2025, 10:14 PM
State	Finished
Completed on	Wednesday, 8 October 2025, 10:37 PM
Time taken	22 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

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 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
#include<stdio.h>
#include<stdlib.h>

int n;

void longestSubsequence(int (*arr)){
    if(n<=2){
        printf("%d\n",n);
        return;
    }

    int (*dp)=malloc(sizeof(int)*n);

    for (int i = 0; i < n; i++) dp[i] = 1;

    for (int i = 1; i < n; i++) {
        for (int j = 0; j < i; j++) {
            if (arr[j] <= arr[i] && dp[i] < dp[j] + 1) {
```

	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 29
State	Finished
Completed on	Wednesday, 29 October 2025, 8:35 AM
Time taken	8 mins 42 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

undefined ×

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 2 3 4	1

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<stdlib.h>

int main(){
    int n;
    scanf("%d",&n);
    int *a=(int*)malloc(n*sizeof(int));

    for(int i=0;i<n;i++) scanf("%d",(a+i));

    int *hash = (int *)calloc(n, sizeof(int)); // initialize with 0s
    if (hash == NULL) {
        printf("Memory allocation failed\n");
        free(a);
        return 1;
    }

    int duplicateFound = 0;
    for (int i = 0; i < n; i++) {
        if (hash[a[i]] == 1) {
            printf("%d ", a[i]);
            duplicateFound = 1;
        } else {
            hash[a[i]]++;
        }
    }

    if (!duplicateFound)
        printf("No duplicates found.");

    free(a);
    free(hash);

    return 0;
}

```

	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 29
State	Finished
Completed on	Wednesday, 29 October 2025, 8:37 AM
Time taken	1 min 15 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

undefined ×

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 2 3 4	1

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    int a[n];
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);

    // Floyd's cycle detection
    int slow = a[0];
    int fast = a[0];

    // Step 1: Move slow by 1 step, fast by 2 steps until they meet
    do {
        slow = a[slow];
        fast = a[a[fast]];
    } while (slow != fast);

    // Step 2: Find entry point of cycle
    slow = a[0];
    while (slow != fast) {
        slow = a[slow];
        fast = a[fast];
    }

    printf("%d", slow);
    return 0;
}
```

	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	

	Input	Expected	Got	
	5 1 1 2 3 4	1	1	

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Back to Course

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 29
State	Finished
Completed on	Wednesday, 29 October 2025, 8:51 AM
Time taken	14 mins 16 secs
Marks	1.00/1.00
Grade	**30.00** out of 30.00 (**100**%)

undefined ×

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:

1. Line 1 contains N1, followed by N1 integers of the first array
2. 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 3 10 17 57 6 2 7 10 15 57 246	10 57

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int T;
    scanf("%d",&T);
    while(T--){
        int n, m;
        scanf("%d", &n);

        // printf("%d %d\n",n,m);
        int *a = malloc(sizeof(int) * n);

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

```c
        {scanf("%d", a + i); /*printf("%d ",*(a+i));*/} //printf("

        scanf("%d", &m);

        int *b = malloc(sizeof(int) * m);

        for (int i = 0; i < m; i++)
            {scanf("%d", b + i); /*printf("%d ",*(b+i));*/}

        // ---- Intersection logic ----
        int found = 0;

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                if (a[i] == b[j]) {
                    printf("%d ", a[i]);
                    found = 1;
                    break;  // avoid printing duplicates for same a[i]
                }
            }
        }

        if (!found)
            printf("No intersection");

        printf("\n");

        // free memory
        free(a);
        free(b);
    }

    return 0;
}
```

	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

undefined ×

Started on	Wednesday, 29
State	Finished
Completed on	Wednesday, 29 October 2025, 8:53 AM
Time taken	1 min 29 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:

1. Line 1 contains N1, followed by N1 integers of the first array
2. 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 3 10 17 57 6 2 7 10 15 57 246	10 57

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int T;
    scanf("%d", &T);  // number of test cases

    while (T--) {
        int n1, n2;

        // Read first array
        scanf("%d", &n1);
        int *a = malloc(n1 * sizeof(int));
        for (int i = 0; i < n1; i++)
```

```c
        scanf("%d", &a[i]);

    // Read second array
    scanf("%d", &n2);
    int *b = malloc(n2 * sizeof(int));
    for (int i = 0; i < n2; i++)
        scanf("%d", &b[i]);

    // Two-pointer approach
    int i = 0, j = 0;
    int found = 0;

    while (i < n1 && j < n2) {
        if (a[i] == b[j]) {
            printf("%d ", a[i]);
            found = 1;
            i++;
            j++;
        } else if (a[i] < b[j]) {
            i++;
        } else {
            j++;
        }
    }

    if (!found)
        printf("No intersection");

    printf("\n");

    free(a);
    free(b);
  }

  return 0;
}
```

	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 29
State	Finished
Completed on	Wednesday, 29 October 2025, 8:55 AM
Time taken	1 min 30 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

undefined ×

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 != 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 3 5 4	1

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int n;
    scanf("%d", &n);

    int *a = malloc(sizeof(int) * n);
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);

    int k;
    scanf("%d", &k);

    int found = 0;

    // O(n^2) brute force search
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (i != j && a[j] - a[i] == k) {
                found = 1;
                break;
            }
        }
        if (found) break;
    }

    printf("%d", found);

    free(a);
    return 0;
}

```

	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

RAJALAKSHMI ENGINEERING COLLEGE

LOHESH M 2024-CSE

L2

Started on	Wednesday, 29
State	Finished
Completed on	Wednesday, 29 October 2025, 8:56 AM
Time taken	1 min 6 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

undefined ×

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 != 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 3 5 4	1

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int n;
    scanf("%d", &n);

    int *a = malloc(sizeof(int) * n);
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);

    int k;
    scanf("%d", &k);

    int i = 0, j = 1;
    int found = 0;

    while (i < n && j < n) {
        int diff = a[j] - a[i];

        if (i != j && diff == k) {
            found = 1;
            break;
        } else if (diff < k) {
            j++;  // increase difference
        } else {
            i++;  // decrease difference
        }
    }

    printf("%d", found);

    free(a);
    return 0;
```

```
    return 0;
}

```

	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