RAJALAKSHMI ENGINEERING COLLEGE RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

Laboratory Record Notebook

Name : GOWTHAM BR

Register No : 230701524

Branch : B.E COMPUTER SCIENCE AND ENGINEERING

Year : II

Section : C

Semester : III

WEEK 1

FINDING TIME COMPLEXITY OF ALGORITHMS USING COUNTER METHOD

For example:

Input	Result
9	12

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int counter = 0;

void function(int n);

int main() {
    int input = 0;


    scanf("%d", &input);


    function(input);
    counter++;


    printf("%d", counter);

    return 0;
}

void function(int n) {
    int i = 1;
     counter++;

    int s = 1;
    counter++;

    while (s <= n) {
        counter++;
        i++;
        counter++;
        s += i;
        counter++;

    }


}
```

	Input	Expected	Got	
✔	9	12	12	✔
✔	4	9	9	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Jump to...

Problem 2: Finding Complexity using Counter method ►

Answer:

```c
# include <stdio.h>

int counter = 0;

void Factor(int num){
    for(int i = 1;i<= num;i++){
        counter++;
        if(num % i == 0){
            counter++;
        }
        counter++;
    }
}

int main(){
    int n = 0;

    scanf("%d",&n);

    Factor(n);
    counter++;


    printf("%d",counter);
    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.

Finish review

◄ Problem 2: Finding Complexity using Counter method

Jump to...

Problem 4: Finding Complexity using Counter Method ►

Answer:

```c
#include<stdio.h>

int counter = 0;
void function(int n)
{
    int c= 0;

    for(int i=n/2; i<n; i++){
         counter++;

        for(int j=1; j<n; j = 2 * j){
             counter++;


            for(int k=1; k<n; k = k * 2){
                counter++;
                c++;
                 counter++;

        }
        counter++;
    }
    counter++;
}
counter++;
}

int main(){
    int n = 0;

    scanf("%d",&n);

    function(n);
    counter++;


    printf("%d",counter);

    return 0;
}
```

	Input	Expected	Got	
✔	4	30	30	✔
✔	10	212	212	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Answer:

```c
#include <stdio.h>
int count = 0;
void reverse(int n)
{
   int rev = 0, remainder;
   count++;


   while (n != 0)
    {
         count++;
         remainder = n % 10;
         count++;


        rev = rev * 10 + remainder;
        count++;

        n/= 10;
        count++;



    }
    count++;


}
int main(){
    int n ;


    scanf("%d",&n);
    count++;
    reverse(n);
    printf("%d",count);

}
```

	Input	Expected	Got	
✔	12	11	11	✔
✔	1234	19	19	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

WEEK-2

GREEDY ALGORITHMS

Answer: (penalty regime: 0 %)

```c

#include <stdio.h>
#include <limits.h>

int min(int a, int b) {
    return (a < b) ? a : b;
}

int minChange(int V) {
    int denominations[] = {1, 2, 5, 10, 20, 50, 100, 500, 1000};
    int n = sizeof(denominations) / sizeof(denominations[0]);
    int dp[V + 1];
    dp[0] = 0;
    for (int i = 1; i <= V; i++) {
        dp[i] = INT_MAX;
        for (int j = 0; j < n; j++) {
            if (denominations[j] <= i) {
                dp[i] = min(dp[i], dp[i - denominations[j]] + 1);
            }
        }
    }

    return dp[V];
}

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

	Input	Expected	Got	
✔	49	5	5	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

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 findContentChildren(int* g, int gSize, int* s, int sSize) {

    qsort(g, gSize, sizeof(int), compare);
    qsort(s, sSize, sizeof(int), compare);

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


    while (i < gSize && j < sSize) {
        if (s[j] >= g[i])
        j++;
    }{

            satisfied++;
            i++;
        }

    return satisfied;
}

int main() {
    int g[] = {1, 2, 3};
    int s[] = {1, 2};

    int gSize = sizeof(g) / sizeof(g[0]);
    int sSize = sizeof(s) / sizeof(s[0]);

    int result = findContentChildren(g, gSize, s, sSize);
    printf("%d\n", result);
}
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Answer: (penalty regime: 0 %)

```c

#include<stdio.h>
#include<math.h>
int minDistance(int calories[],int n){
    for(int i=0;i<n-1;i++){
        for(int j=0;j<n-i-1;j++){
            if(calories[j]<calories[j+1]){
                int temp=calories[j];
                calories[j]=calories[j+1];
                calories[j+1]=temp;
            }
        }
    }
    int distance=0;
    for(int i=0;i<n;i++){
        distance+=pow(n,i)*calories[i];
    }
    return distance;
}
int main(){
    int n;
    scanf("%d",&n);
    int calories[n];
    for(int i=0;i<n;i++){
        scanf("%d",&calories[i]);
    }
    int min=minDistance(calories,n);
    printf("%d",min);
    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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

void swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = (low - 1);

    for (int j = low; j <= high - 1; j++) {
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }

    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}


void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pivot = partition(arr, low, high);

        quickSort(arr, low, pivot - 1);
        quickSort(arr, pivot + 1, high);
    }
}

// Function to print the array
/*void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
*/
int maximizeSum(int arr[], int n) {
    quickSort(arr, 0,n-1);

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

    return sum;
}
int main() {
```

	Input	Expected	Got	
✔	5 2 5 3 4 0	40	40	✔
✔	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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

void sortAscending(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] > arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

void sortDescending(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] < arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

int minProductSum(int array_One[], int array_Two[], int n) {
     sortAscending(array_One, n);
     sortDescending(array_Two, n);

    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += array_One[i] * array_Two[i];
    }
    return sum;
}

int main() {
    int n;

    scanf("%d", &n);

    int array_One[n], array_Two[n];

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


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

	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.

WEEK-3

DIVIDE AND CONQURE

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>


int countZeros(int arr[], int low, int high) {

    if (low > high)
        return 0;

    if (arr[low] == 0 && arr[high] == 0)
        return high - low + 1;

     if (arr[low] == 1 && arr[high] == 1)
        return 0;

     int mid = (low + high) / 2;

     return countZeros(arr, low, mid) + countZeros(arr, mid + 1, high);
}

int main() {
    int m;


    scanf("%d", &m);

    int arr[m];


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

     int zeroCount = countZeros(arr, 0, m - 1);
    printf( "%d", zeroCount);

    return 0;
}
```

	Input	Expected	Got	
✔	5 1 1 1 0 0	2	2	✔
✔	10 1 1 1 1 1 1 1 1 1 1	0	0	✔
✔	8 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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

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

    int nums[n];

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

    int count = 0;
    int candidate = 0;

    for (int i = 0; i < n; i++) {
        if (count == 0) {
            candidate = nums[i];
        }
        if (nums[i] == candidate) {
            count++;
        } else {
            count--;
        }
    }

    printf("%d\n", candidate);

    return 0;
}
```

	Input	Expected	Got	
✔	3 3 2 3	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n, x;
    scanf("%d", &n);
    int arr[n];

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

    scanf("%d", &x);

    int left = 0, right = n - 1;
    int floor = -1;

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

        if (arr[mid] == x) {
            floor = arr[mid];
            break;
        }

        if (arr[mid] < x) {
            floor = arr[mid];
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }

    if (floor != -1) {
        printf("%d\n", floor);
    } else {
        printf("No floor value found for %d in the array.\n", x);
    }

    return 0;
}
```

	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.

Finish review

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

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

    scanf("%d", &x);

    int left = 0, right = n - 1;
    int found = 0;

    while (left < right) {
        int sum = arr[left] + arr[right];

        if (sum == x) {

            printf("%d\n", arr[left]);
            printf("%d\n", arr[right]);
            found = 1;
            break;
        }

        if (sum < x) {
            left++;
        } else {
            right--;
        }
    }

    if (!found) {
        printf("No");
    }

    return 0;
}
```

	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.

Answer:

```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]);
    }


    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (a[j] < a[i]) {
                // Swap a[i] and a[j]
                int temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }


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

    return 0;
}

```

	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.

WEEK-4

DYNAMIC PROGRAMING

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

long long count_ways(int n) {

    long long dp[n + 1];

    dp[0] = 1;
    dp[1] = 1;
    dp[2] = 1;
    dp[3] = 2;


    for (int i = 4; i <= n; i++) {
        dp[i] = dp[i - 1] + dp[i - 3];
    }

    return dp[n];
}

int main() {
    int n;
    scanf("%d", &n);
    printf("%lld\n", count_ways(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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#define MAX 100


int max(int a, int b) {
    return (a > b) ? a : b;
}


int maxMonetaryPath(int chessboard[MAX][MAX], int n) {
    int dp[MAX][MAX];


    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            dp[i][j] = 0;
        }
    }


    dp[0][0] = chessboard[0][0];


    for (int j = 1; j < n; j++) {
        dp[0][j] = dp[0][j - 1] + chessboard[0][j];
    }


    for (int i = 1; i < n; i++) {
        dp[i][0] = dp[i - 1][0] + chessboard[i][0];
    }

    for (int i = 1; i < n; i++) {
        for (int j = 1; j < n; j++) {
            dp[i][j] = chessboard[i][j] + max(dp[i - 1][j], dp[i][j - 1]);
        }
    }


    return dp[n - 1][n - 1];
}

int main() {
    int n;
    int chessboard[MAX][MAX];


    scanf("%d", &n);


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

	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.

Answer: (penalty regime: 0 %)

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


int longest_common_subsequence(char s1[], char s2[]) {
    int m = strlen(s1);
    int n = strlen(s2);
    int dp[m + 1][n + 1];


    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0 || j == 0) {
                dp[i][j] = 0;
            } else 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];
            }
        }
    }

    return dp[m][n];
}

int main() {
    char s1[100], s2[100];


    scanf("%s", s1);
    scanf("%s", s2);


    int result = longest_common_subsequence(s1, s2);


    printf("%d", result);

    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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int longest_non_decreasing_subsequence(int arr[], int n) {
    int dp[n];
    int max_length = 1;


    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[i] > dp[j] + 1) ? dp[i] : dp[j] + 1;
            }
        }

        if (dp[i] > max_length) {
            max_length = dp[i];
        }
    }

    return max_length;
}

int main() {
    int n;


    scanf("%d", &n);

    int arr[n];


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


    int result = longest_non_decreasing_subsequence(arr, n);
    printf("%d", result);

    return 0;
}

```

	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.

WEEK-5

COMPITATIVE PROGRAMMING

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>


int findDuplicate(int arr[], int n) {
    int slow = arr[0];
    int fast = arr[arr[0]];


    while (slow != fast) {
        slow = arr[slow];
        fast = arr[arr[fast]];
    }


    fast = 0;
    while (slow != fast) {
        slow = arr[slow];
        fast = arr[fast];
    }

    return slow;
}

int main() {
    int n;


    scanf("%d", &n);

    int arr[n];


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


    int duplicate = findDuplicate(arr, n);
    printf("%d\n", duplicate);

    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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int findDuplicate(int arr[], int n) {
    int slow = arr[0];
    int fast = arr[arr[0]];

    // Phase 1: Detect cycle
    while (slow != fast) {
        slow = arr[slow];
        fast = arr[arr[fast]];
    }

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

    return slow;
}

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

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

    int duplicate = findDuplicate(arr, n);
    printf("%d", duplicate);

    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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

void findIntersection(int arr1[], int n1, int arr2[], int n2) {
    int i = 0, j = 0;
    int found = 0;

    while (i < n1 && j < n2) {
        if (arr1[i] < arr2[j]) {
            i++;
        } else if (arr2[j] < arr1[i]) {
            j++;
        } else {
            // Intersection found
            printf("%d ", arr1[i]);
            found = 1;
            i++;
            j++;
        }
    }

    if (!found) {
        printf("No Intersection");
    }
    printf("\n");
}

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

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


        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++) {
            scanf("%d", &arr1[i]);
        }


        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 0; i < n2; i++) {
            scanf("%d", &arr2[i]);
        }


        findIntersection(arr1, n1, arr2, n2);
    }

    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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

void findIntersection(int arr1[], int n1, int arr2[], int n2) {
    int i = 0, j = 0;
    int found = 0;

    while (i < n1 && j < n2) {
        if (arr1[i] < arr2[j]) {
            i++;
        } else if (arr2[j] < arr1[i]) {
            j++;
        } else {

            printf("%d ", arr1[i]);
            found = 1;
            i++;
            j++;
        }
    }

    if (!found) {
        printf("No Intersection");
    }
    printf("\n");
}

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

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


        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++) {
            scanf("%d", &arr1[i]);
        }


        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 0; i < n2; i++) {
            scanf("%d", &arr2[i]);
        }


        findIntersection(arr1, n1, arr2, n2);
    }

    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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int findPairWithDifference(int arr[], int n, int k) {
    int i = 0, j = 1;

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

        if (i != j && diff == k) {

            return 1;
        }
        else if (diff < k) {
            j++;
        }
        else {
            i++;
        }
    }

    return 0;
}

int main() {
    int n, k;
    scanf("%d", &n);
    int arr[n];

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

    scanf("%d", &k);
    int result = findPairWithDifference(arr, n, k);
    printf("%d\n", result);

    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.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int findPairWithDifference(int arr[], int n, int k) {
    int i = 0, j = 1;

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

        if (i != j && diff == k) {

            return 1;
        }
        else if (diff < k) {

            j++;
        }
        else {

            i++;

            if (i == j) {
                j++;
            }
        }
    }

    return 0;
}

int main() {
    int n, k;
    scanf("%d", &n);
    int arr[n];

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

    scanf("%d", &k);
    int result = findPairWithDifference(arr, n, k);
    printf("%d", result);

    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.