You have already solved this challenge ! Though you can run the code with different logic !

Code Editor

Code Editor | GCC v6.3.0 | Light Theme

```c
#include <stdio.h>
#include <string.h>
void check_subsequence(char a[],char b[]){
 int c=0,d=0;
 while(a[c]!='\0'){
 while(a[c]!=b[d]&& b[d]!='\0')
 d++;
 if(b[d]=='\0')
 break;
 d++;c++;
}
(a[c] =='\0')?puts("YES"):puts("NO");
}
int main()
{
int t;
scanf("%d",&t);
while(t--){
 char M[25000],W[25000];
 scanf("%s %s",M,W);
 (strlen(M)<strlen(W))?check_subsequence(M,W):check_subsequence(W,M);
}
return 0;
}
```

Save | Reset | Run | Evaluate

Congratulations

You solved this challenge

Custom Input (stdin) T1 | T2

Type Here

Output Match T1 | Match T2

Successfully Executed !

Complexity Analysis

Cyclomatic Complexity : 6
Token Count : 170
NLOC : 24
Execution Time : 132 ms
Size : 430 bytes

Test Case Status

- Logical T1 Passed
- Logical T2 Passed
- Logical T3 Passed
- Logical T4 Passed
- Mandatory T1 Passed
- Mandatory T2 Passed
- Mandatory T3 Passed
- Mandatory T4 Passed
- Complexity T1 Passed
- Complexity T2 Passed
- Complexity T3 Passed

You have already solved this challenge ! Though you can run the code with different logic !

Code Editor

Code Editor | GCC v6.3.0 | Light Theme

```c
#include <stdio.h>
void find_index(int arr[100],int n,int budget,int *ans1,int *ans2)
{ int i,j;
 for(i=1;i<=n;i++)
 { for(j=i+1;j<=n;j++)
 { if(arr[i]+arr[j]==budget)
 { *ans1=i;
 *ans2=j;
 }
 }
 }
}
int main()
{ int n,i,ans1,ans2,t,budget,a[100];
 int *arr=a;
 scanf("%d",&t);
 while(t--)
 { scanf("%d",&budget);
 scanf("%d",&n);
 for(i=1;i<=n;i++)
 scanf("%d",(arr+i));
 find_index(a,n,budget,&ans1,&ans2);
 printf("%d %d\n",ans1,ans2);
 }
return 0;
}
```

Save | Reset | Run | Evaluate

Congratulations

You solved this challenge

Custom Input (stdin) T1 T2

Type Here

Output Match T1 Match T2

```
Successfully Executed !
```

Complexity Analysis

Cyclomatic Complexity : 4
Token Count : 202
NLOC : 26
Execution Time : 148 ms
Size : 480 bytes

Test Case Status

Logical T1 Passed	Logical T2 Passed
Logical T3 Passed	Logical T4 Passed
Mandatory T1 Passed	Mandatory T2 Passed
Mandatory T3 Passed	Complexity T1 Passed
Complexity T2 Passed	Complexity T3 Passed

Tuples

Problem | Submissions | Leaderboard | Discussions

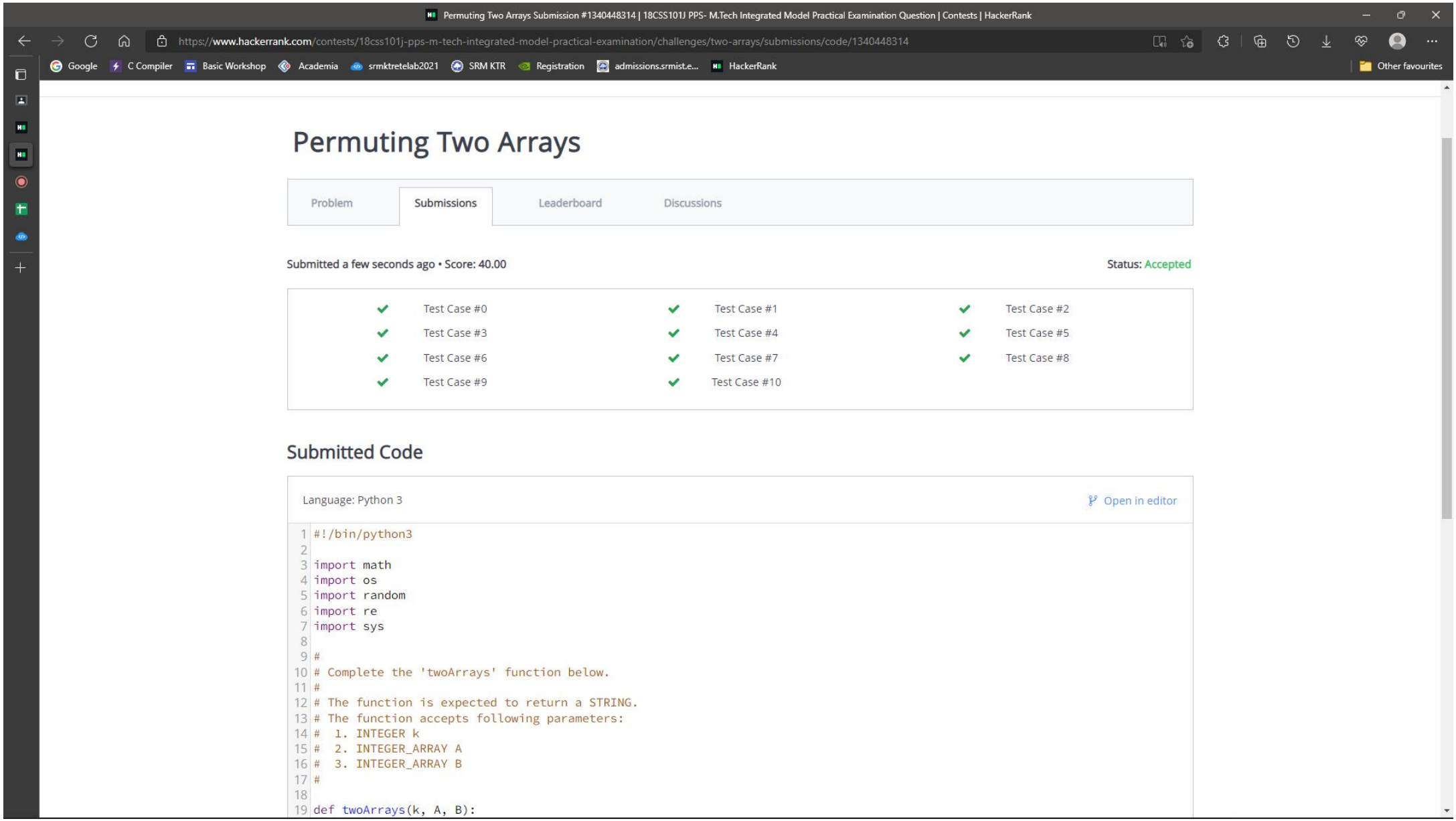
Submitted a few seconds ago • Score: 10.00 Status: Accepted

✔ Test Case #0 ✔ Test Case #1

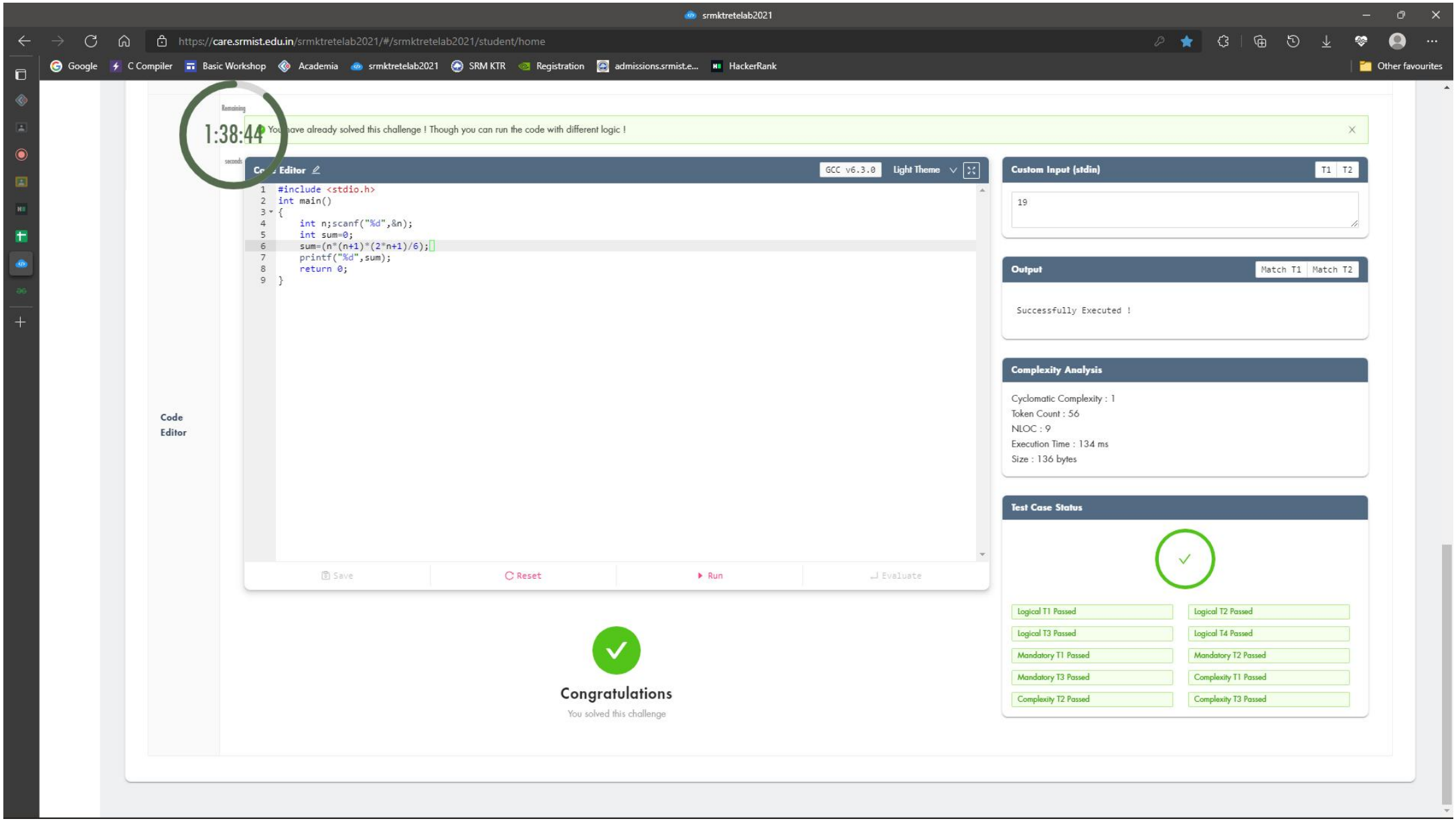
Submitted Code

Language: Python 2 Open in editor

```
1 if __name__ == '__main__':
2     n = int(raw_input())
3     integer_list = map(int, raw_input().split())
4     t = tuple(integer_list)
5     print(hash(t))
6
```


Permuting Two Arrays Submission #1340448314 | 18CSS101J PPS- M.Tech Integrated Model Practical Examination Question | Contests | HackerRank
https://www.hackerrank.com/contests/18css101j-pps-m-tech-integrated-model-practical-examination/challenges/two-arrays/submissions/code/1340448314
Google
C Compiler
Basic Workshop
Academia
srmktretelab2021
SRM KTR
Registration
admissions.srmist.e...
HackerRank
Other favourites
Permuting Two Arrays
Problem
Submissions
Leaderboard
Discussions
Submitted a few seconds ago • Score: 40.00
Status: Accepted
Test Case #0
Test Case #1
Test Case #2
Test Case #3
Test Case #4
Test Case #5
Test Case #6
Test Case #7
Test Case #8
Test Case #9
Test Case #10
Submitted Code
Language: Python 3
Open in editor
#!/bin/python3
import math
import os
import random
import re
import sys
#
Complete the 'twoArrays' function below.
#
The function is expected to return a STRING.
The function accepts following parameters:
1. INTEGER k
2. INTEGER_ARRAY A
3. INTEGER_ARRAY B
#
def twoArrays(k, A, B):

```python

#
# Complete the 'twoArrays' function below.
#
# The function is expected to return a STRING.
# The function accepts following parameters:
#  1. INTEGER k
#  2. INTEGER_ARRAY A
#  3. INTEGER_ARRAY B
#

def twoArrays(k, A, B):
    n = len(A)
    A.sort()
    B.sort(reverse=True)
    for i in range(n):
        if A[i]+B[i] < k:
            return "NO"
    return "YES"


if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    q = int(input().strip())

    for q_itr in range(q):
        first_multiple_input = input().rstrip().split()

        n = int(first_multiple_input[0])

        k = int(first_multiple_input[1])

        A = list(map(int, input().rstrip().split()))

        B = list(map(int, input().rstrip().split()))

        result = twoArrays(k, A, B)

        fptr.write(result + '\n')

    fptr.close()

```


srmktretelab2021
https://care.srmist.edu.in/srmktretelab2021/#/srmktretelab2021/student/home
Google
C Compiler
Basic Workshop
Academia
srmktretelab2021
SRM KTR
Registration
admissions.srmist.e...
HackerRank
Other favourites
Remaining
1:38:44
seconds
You have already solved this challenge ! Though you can run the code with different logic !
Code Editor
GCC v6.3.0
Light Theme
#include <stdio.h>
int main()
{
int n;scanf("%d",&n);
int sum=0;
sum=(n*(n+1)*(2*n+1)/6);
printf("%d",sum);
return 0;
}
Save
Reset
Run
Evaluate
Code Editor
Custom Input (stdin)
T1
T2
19
Output
Match T1
Match T2
Successfully Executed !
Complexity Analysis
Cyclomatic Complexity : 1
Token Count : 56
NLOC : 9
Execution Time : 134 ms
Size : 136 bytes
Test Case Status
Logical T1 Passed
Logical T2 Passed
Logical T3 Passed
Logical T4 Passed
Mandatory T1 Passed
Mandatory T2 Passed
Mandatory T3 Passed
Complexity T1 Passed
Complexity T2 Passed
Complexity T3 Passed
Congratulations
You solved this challenge

role student | name aniruddha | ID 465545688193 | dept school of computing | January 6th 2022, 9:15:45 am | Logout

Test mode Activated

User ID
465545688193

Name
ANIRUDDHA

Test Available
1

Assessment will closed & get auto logout, when times up !

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	C	Session	Input & Output	Question Information	Level 1 • Challenge 3

Problem

Problem Description:

Tina's brother gave her a friendly task of calculating the number of squares in a board that has n*n squares of dimensions 1cm *1cm each.

Help her to find the number of total squares including all small and big ones.

Constraints:

$2 \leq n \leq 20$

Input Format:

The only line of the input represents a value of "n"

Output Format:

Print the number of squares in the n*n board."

Test Cases

Logical Test Cases

Test Case 1
INPUT (STDIN): 14
EXPECTED OUTPUT: 1015

Test Case 2
INPUT (STDIN): 19
EXPECTED OUTPUT: 2470

Mandatory Test Cases

Test Case 1
KEYWORD: int n;

Test Case 2
KEYWORD: scanf

Test Case 3
KEYWORD: printf

Complexity Test Cases

Test Case 1
CYCLOMATIC COMPLEXITY: 1

Test Case 2
TOKEN COUNT: 65

Test Case 3
NLOC: 9

Code Editor

Remaining 1:39:22 seconds

You have already solved this challenge ! Though you can run the code with different logic !

Code Editor — GCC v6.3.0 — Light Theme

```
1  #include <stdio.h>
2  int main()
3  {
4      int n;scanf("%d",&n);
5      int sum=0;
6      sum=(n*(n+1)*(2*n+1)/6);
7      printf("%d",sum);
8      return 0;
9  }
```

Save | Reset | Run | Evaluate

Custom Input (stdin) — T1 | T2

19

Output — Match T1 | Match T2

Successfully Executed !

Complexity Analysis

Cyclomatic Complexity : 1
Token Count : 56
NLOC : 9
Execution Time : 134 ms
Size : 136 bytes

Test Case Status

Logical T1 Passed | Logical T2 Passed
Logical T3 Passed | Logical T4 Passed
Mandatory T1 Passed | Mandatory T2 Passed
Mandatory T3 Passed | Complexity T1 Passed
Complexity T2 Passed | Complexity T3 Passed

Congratulations
You solved this challenge