# DAA SKILL

## WEEK 9

**A. Neeharika**

**2300030030**

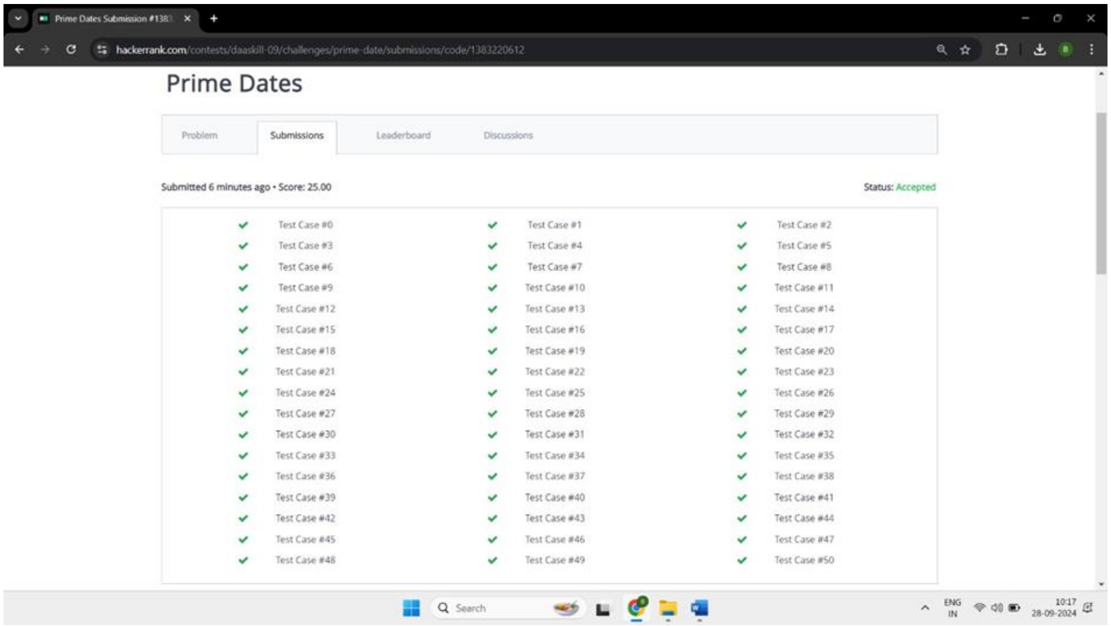
**PRIME DATES**

**PROGRAM:**

```
import re
month = []
def updateLeapYear(year):
    if year % 400 == 0:
        month[2] = 29
    elif year % 100 == 0:
        month[2] = 28
    elif year % 4 == 0:
        month[2] = 29
    else:
        month[2] = 28
def storeMonth():
    month[1] = 31
    month[2] = 28
    month[3] = 31
    month[4] = 30
    month[5] = 31
    month[6] = 30
    month[7] = 31

    month[8] = 31

    month[9] = 30

    month[10] = 31

    month[11] = 30

    month[12] = 31

   def findPrimeDates(d1, m1, y1, d2, m2, y2): storeMonth()

    result = 0

    while(True): x = d1

    x = x * 100 + m1 x = x * 10000 + y1

    if x % 4 == 0 or x % 7 == 0: result = result + 1

    if d1 == d2 and m1 == m2 and y1 == y2: break

    updateLeapYear(y1) d1 = d1 + 1

    if d1 > month[m1]: m1 = m1 + 1

    d1 = 1

    if m1 > 12:

    y1 = y1 + 1 m1 = 1

    return result;

    for i in range(1, 15): month.append(31)

    line = input()

    date = re.split('-| ', line)
d1 = int(date[0])

m1 = int(date[1])

y1 = int(date[2])

d2 = int(date[3])

m2 = int(date[4])
```

```
y2 = int(date[5])
result = findPrimeDates(d1, m1, y1, d2, m2, y2)
print(result)
```

**Output:**



Prime Dates Submission #1383
hackerrank.com/contests/daaskill-09/challenges/prime-date/submissions/code/1383220612

## Prime Dates

Problem
Submissions
Leaderboard
Discussions

Submitted 6 minutes ago • Score: 25.00

Status: Accepted

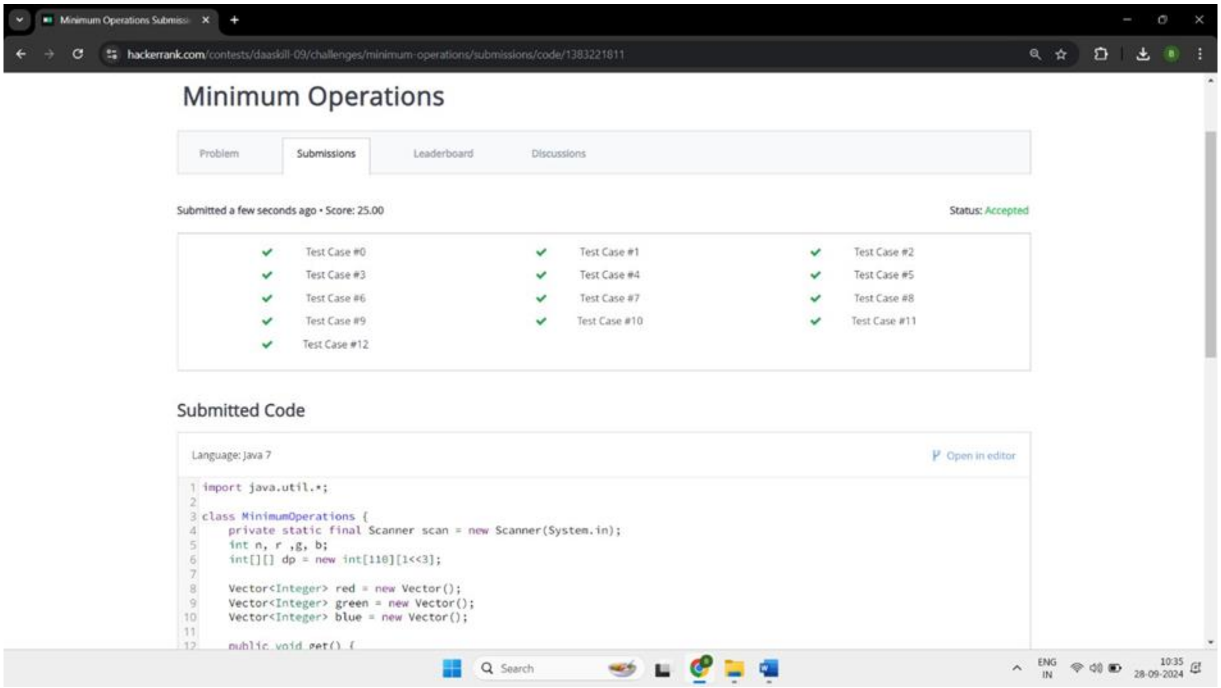
| | | | | | |
|---|---|---|---|---|---|
| ✓ | Test Case #0 | ✓ | Test Case #1 | ✓ | Test Case #2 |
| ✓ | Test Case #3 | ✓ | Test Case #4 | ✓ | Test Case #5 |
| ✓ | Test Case #6 | ✓ | Test Case #7 | ✓ | Test Case #8 |
| ✓ | Test Case #9 | ✓ | Test Case #10 | ✓ | Test Case #11 |
| ✓ | Test Case #12 | ✓ | Test Case #13 | ✓ | Test Case #14 |
| ✓ | Test Case #15 | ✓ | Test Case #16 | ✓ | Test Case #17 |
| ✓ | Test Case #18 | ✓ | Test Case #19 | ✓ | Test Case #20 |
| ✓ | Test Case #21 | ✓ | Test Case #22 | ✓ | Test Case #23 |
| ✓ | Test Case #24 | ✓ | Test Case #25 | ✓ | Test Case #26 |
| ✓ | Test Case #27 | ✓ | Test Case #28 | ✓ | Test Case #29 |
| ✓ | Test Case #30 | ✓ | Test Case #31 | ✓ | Test Case #32 |
| ✓ | Test Case #33 | ✓ | Test Case #34 | ✓ | Test Case #35 |
| ✓ | Test Case #36 | ✓ | Test Case #37 | ✓ | Test Case #38 |
| ✓ | Test Case #39 | ✓ | Test Case #40 | ✓ | Test Case #41 |
| ✓ | Test Case #42 | ✓ | Test Case #43 | ✓ | Test Case #44 |
| ✓ | Test Case #45 | ✓ | Test Case #46 | ✓ | Test Case #47 |
| ✓ | Test Case #48 | ✓ | Test Case #49 | ✓ | Test Case #50 |

**2.MINIMUM OPERATIONS**

**PROGRAM:**

```
import java.util.*;

class MinimumOperations {
private static final Scanner scan = new Scanner(System.in);
int n, r ,g, b;
int[][] dp = new int[110][1<<3];

Vector<Integer> red = new Vector();
Vector<Integer> green = new Vector();
Vector<Integer> blue = new Vector();

public void get() {
n = scan.nextInt();

for (int i = 0; i < n; i++) {
r = scan.nextInt();
g  = scan.nextInt();
b  = scan.nextInt();
red.add(r);
green.add(g);
blue.add(b);
}

}


public void minOperations() {
int i, j;
for (i = 0; i <= n; i++) {
for (j = 0; j <= 7; j++) {
dp[i][j] = (1<<30);
}

}

dp[0][0] = 0;
for (i = 0; i < n; i++){
for (j = 0; j <= 7; j++){
dp[i + 1][j | 1] = Math.min(dp[i + 1][j | 1], dp[i][j] + green.get(i) + blue.get(i));
dp[i + 1][j | 2] = Math.min(dp[i + 1][j | 2], dp[i][j] + red.get(i) + blue.get(i));
dp[i + 1][j | 4] = Math.min(dp[i + 1][j | 4], dp[i][j] + red.get(i) + green.get(i));
}

}

j = 0;
for (i = 0; i < n; i++){
if (green.get(i) != 0) j |= 1;
if (red.get(i) != 0) j |= 2;
if (blue.get(i) != 0) j |= 4;
}

if (dp[n][j] >= (1<<30)) dp[n][j] = -1;
System.out.println(dp[n][j]);
}

}

class Solution {
public static void main(String[] args) {
MinimumOperations obj = new MinimumOperations();
obj.get();
obj.minOperations();
}

}
```

# OUTPUT:


Minimum Operations Submissi
hackerrank.com/contests/daaskill-09/challenges/minimum-operations/submissions/code/1383221811
Problem
Submissions
Leaderboard
Discussions
Open in editor
Search
ENG
IN
10:35
28-09-2024

## Minimum Operations

Submitted a few seconds ago • Score: 25.00

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 ✓ Test Case #11
✓ Test Case #12

## Submitted Code

Language: Java 7

```java
import java.util.*;

class MinimumOperations {
    private static final Scanner scan = new Scanner(System.in);
    int n, r ,g, b;
    int[][] dp = new int[110][1<<3];

    Vector<Integer> red = new Vector();
    Vector<Integer> green = new Vector();
    Vector<Integer> blue = new Vector();

    public void get() {
```