Experiment 9

Student Name: Ankita **UID:** 22BCS17162

Branch: CSE **Section/Group:**IOT_NTPP_602-A

Semester: 6th **Date of Performance:**17-04-25

Subject Name: AP- 2 **Subject Code:** 22CSP-351

Aim:

a) Hamming Disatance
b) Task Schedular
c) Divide two integers

Objective: To learn some miscellaneous problems.

Code:

a)

```
class Solution {
public:
int hammingDistance(int x, int y) {
int xorResult = x ^ y;
int count = 0;
while (xorResult) {
count += xorResult & 1;
xorResult >>= 1;
}
return count;
}};
```

b)

```
class Solution {
public:
int leastInterval(vector<char>& tasks, int n) {
vector<int> freq(26, 0);
```

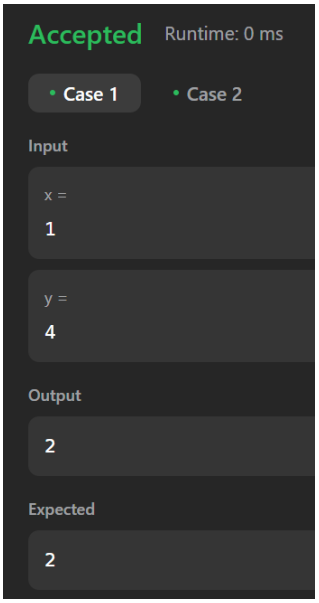
```
for (char task : tasks) {
freq[task - 'A']++;
}
sort(freq.begin(), freq.end());
int maxFreq = freq[25];
int idleSlots = (maxFreq - 1) * n;
for (int i = 24; i >= 0 && idleSlots > 0; --i) {
idleSlots -= min(freq[i], maxFreq - 1);
}
idleSlots = max(0, idleSlots);
return tasks.size() + idleSlots;
}
};
```

c)

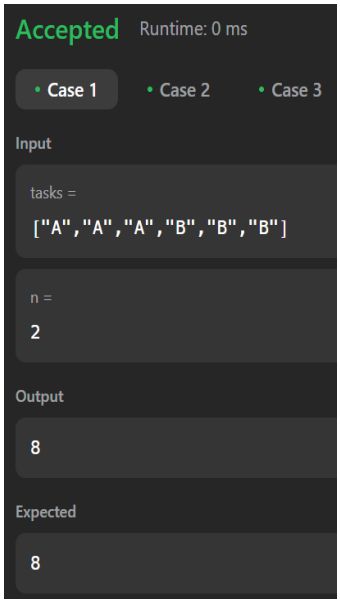
```
class Solution {
public:
int divide(int dividend, int divisor) {
if (dividend == INT_MIN && divisor == -1)
return INT_MAX;
long long a = abs((long long)dividend);
long long b = abs((long long)divisor);
int result = 0;
while (a >= b) {
long long temp = b, multiple = 1;
while (a >= (temp << 1)) {
temp <<= 1;
multiple <<= 1;
}
a -= temp;
result += multiple;
}
if ((dividend < 0) ^ (divisor < 0)) {
return -result;
} else {
return result;
}
}};
```

Output:

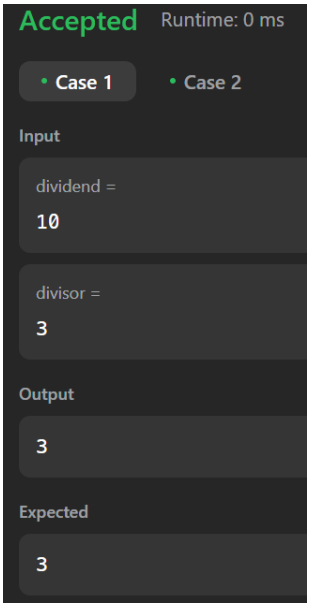
a)



b)



c)



Learning Outcomes:

a) Understand the concept of graph.
b) Learnt about different problem like word ladder, number of islands, etc.
c) Gain an understanding about the efficiency of graph.