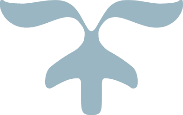


DAA WEEK – 1 SKILL – 1



# [2D Array - DS](https://www.hackerrank.com/contests/evensem-daa-skill-01/challenges/2d-array)

#include <stdio.h>

int hourglassSum(int arr[6][6]) {

int max = -100, sum;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

sum = arr[i][j] + arr[i][j+1] + arr[i][j+2] +

arr[i+1][j+1] +

arr[i+2][j] + arr[i+2][j+1] + arr[i+2][j+2];

if (sum > max) max = sum;

}

}

return max;

}

int main() {

int arr[6][6];

for (int i = 0; i < 6; i++)

for (int j = 0; j < 6; j++)

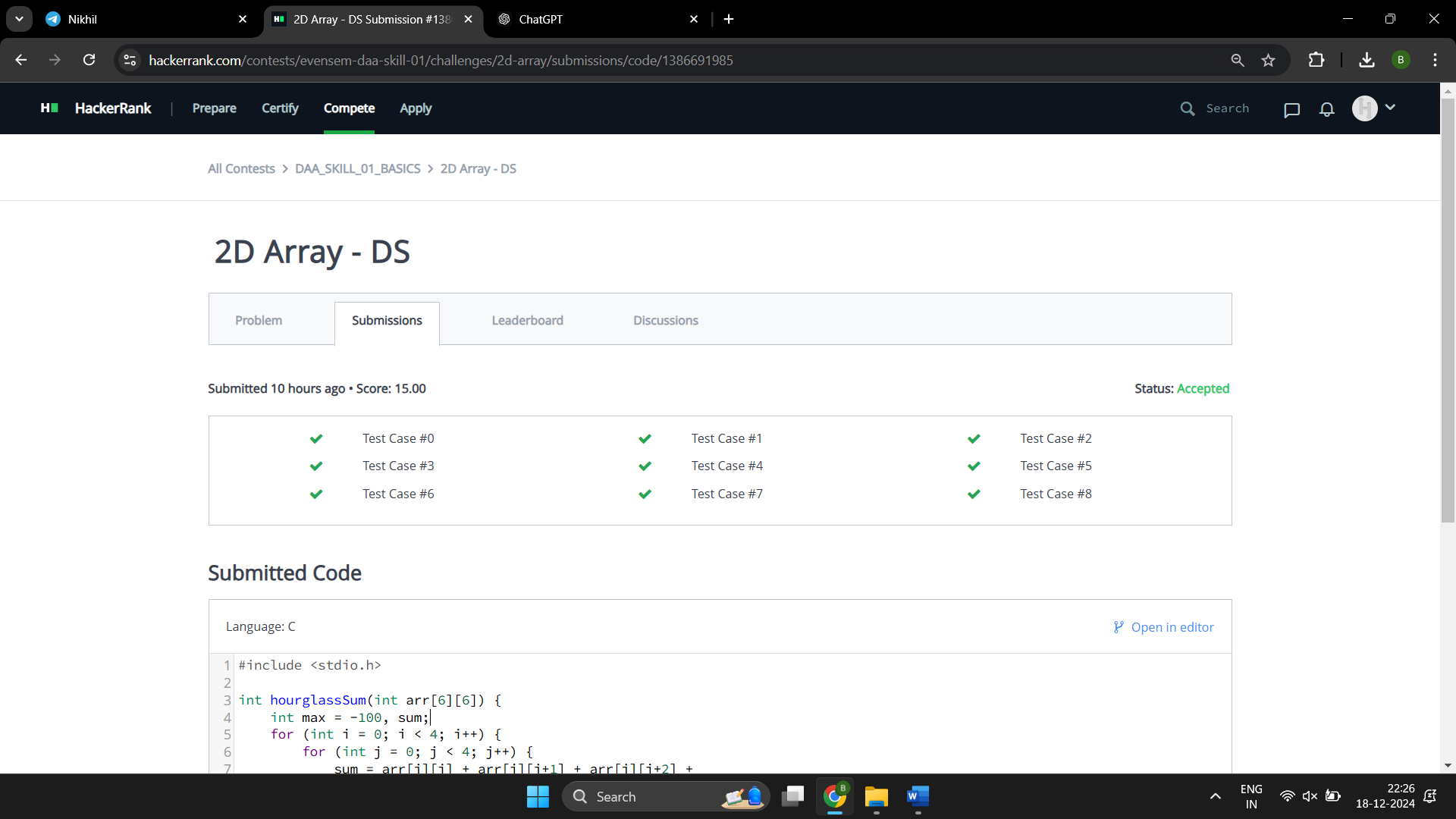
scanf("%d", &arr[i][j]);

printf("%d\n", hourglassSum(arr));

return 0;

}

**2D Array - DS Test Cases**

****

# [Dynamic Array](https://www.hackerrank.com/contests/evensem-daa-skill-01/challenges/dynamic-array)

#include <stdio.h>

#include <stdlib.h>

int main() {

int n, q, t, x, y, lastAnswer = 0;

scanf("%d %d", &n, &q);

int\*\* seqList = (int\*\*)calloc(n, sizeof(int\*));

int\* sizes = (int\*)calloc(n, sizeof(int));

while (q--) {

scanf("%d %d %d", &t, &x, &y);

int idx = (lastAnswer ^ x) % n;

if (t == 1) {

seqList[idx] = (int\*)realloc(seqList[idx], ++sizes[idx] \* sizeof(int));

seqList[idx][sizes[idx] - 1] = y;

} else {

lastAnswer = seqList[idx][y % sizes[idx]];

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

}

}

for (int i = 0; i < n; i++) free(seqList[i]);

free(seqList);

free(sizes);

return 0;

}

**Dynamic Array Test Cases**

**A screenshot of a computer

Description automatically generated**

# [Left Rotation](https://www.hackerrank.com/contests/evensem-daa-skill-01/challenges/array-left-rotation)

#include<stdio.h>

int main()

{

int n, m, temp, arr[100], i;

scanf("%d%d", &n, &m);

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

{

temp = (n + i - m) % n;

scanf("%d", &arr[temp]);

}

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

{

printf("%d ", arr[i]);

}

}

**Left Rotation Test Cases**

**A screenshot of a computer

Description automatically generated**

# [Sparse Arrays](https://www.hackerrank.com/contests/evensem-daa-skill-01/challenges/sparse-arrays)

#include <stdio.h>

#include <string.h>

#define MAX\_STRINGS 1000

#define MAX\_QUERIES 100

int main() {

int n, q;

char str[MAX\_STRINGS][100], query[100];

scanf("%d", &n);

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

scanf("%s", str[i]);

}

scanf("%d", &q);

while (q--) {

scanf("%s", query);

int count = 0;

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

if (strcmp(str[i], query) == 0) count++;

}

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

}

return 0;

}

**Sparse Arrays Test Cases**

**A screenshot of a computer

Description automatically generated**

# [Print in Reverse](https://www.hackerrank.com/contests/evensem-daa-skill-01/challenges/print-the-elements-of-a-linked-list-in-reverse)

void reversePrint(SinglyLinkedListNode\* head)

{

if (head == NULL) return;

reversePrint(head->next);

printf("%d\n", head->data);

}

**Print in Reverse Test Cases**

**A screenshot of a computer

Description automatically generated**