

Worksheet 4

Name: Anirudh Bhardwaj

Class: DWWC 43

UID: 20BCS2540

Subject: DSA

1. Aim/Overview of the practical:

- i) You are participating in a contest which has 1111 problems (numbered 11 through 1111). The first eight problems (i.e. problems 1, 2, ..., 8) are *scorable*, while the last three problems (99, 1010 and 1111) are *nonscorable* — this means that any submissions you make on any of these problems do not affect your total score.

Your total score is the sum of your best scores for all scorable problems. That is, for each scorable problem, you look at the scores of all submissions you made on that problem and take the maximum of these scores (or 00 if you didn't make any submissions on that problem); the total score is the sum of the maximum scores you took.

You know the results of all submissions you made. Calculate your total score.

- ii) Shahid is a computer science student and his teacher gave him a simple question to solve within time but he is little busy in his life, so he asked you to solve this problem.

iii)

A team of young programmers was playing with the contents of a two-dimensional matrix in a systematic manner and suddenly team members found an interesting thing. They observed that the contents are getting arranged in an increasing order when one moves either in a row (left to right) or in a column (top to bottom). Thus they decided to implement the systematic procedure which they followed while playing. They developed a document first and then share it with other teams too so that they can also verify the said observation.

Problem statement : You have been given an array of positive integers A_1, A_2, \dots, A_n with length N and you have to print an array of same length(N) where the values in the new array are the sum of every number in the array, except the number at that index.

Here's what they shared:

- Let A be a square $n \times n$ matrix of integers.
- Rows/columns with lower indices have to be processed first.
- The process has to be repeated alternatively on rows and columns.
- Overall the processing sequence to be followed is $row_0, column_0, row_1, column_1, \dots, row_{n-1}, column_{n-1}$.
- If we are at i^{th} row, then we have to work with each column at a time from 0 to $n-1$ of this row. For any j^{th} column, swap $A[i][j]$ with the minimum of all the elements which are present in a column with index j and rows from indices i to $n-1$.
- If we are at j^{th} column, then we have to work with each row at a time from 0 to $n-1$ of this column. For any i^{th} row, swap $A[i][j]$ with the minimum of all the elements which are present in a row with index i and columns with indices j to $n-1$.

Let you, being a member of one of the senior teams, received the same. To do a bit of analysis, you decided to proceed with the implementation and also to keep a count on the total number of swaps.

###Input:

- Line 1 contains an integer N , the size of the square matrix.
- Line 2 contains $N \times N$ integers separated by space. These are the contents of a square matrix in row-major order.

###Output:

- Line 1 is an integer giving the total number of swaps.
- Line 2 is space separated $N \times N$ integers sequence. These are the final contents of a square matrix in row-major order.

###Sample Input: 3 19 28 39 21 2 11 22 12 37

###Sample Output: 8 2 11 19 12 22 37 21 28 39

###EXPLANATION: The array contents (listed in row-major order) get updated in the following manner after processing each row/column completely:

19 28 39 21 2 11 22 12 37

19 2 11 21 28 39 22 12 37

2 19 11 21 28 39 12 22 37

2 19 11 12 22 37 21 28 39

2 11 19 12 22 37 21 28 39

For example, if $n = 3n=3$, then $LL1: 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow NIL$ $LL2:$

$4 \rightarrow 8 \rightarrow NIL$ $LL3: 6 \rightarrow 10 \rightarrow NIL$

$LL3: 6 \rightarrow 10 \rightarrow NIL$

Now combine $LL1LL1$ with $LL2LL2$ and $LL3LL3$. Nodes of $LL2LL2$ are to be placed at alternative positions in the first-half of $LL1LL1$ and nodes of $LL3LL3$ are to be placed at alternative positions in the last-half of $LL1LL1$. Create a new node $MIDMID$ that contains sum of first and last node values of $LL1LL1$ and place it in the middle of the Updated $LL1UpdatedLL1$.

Note: Creation of new node is not allowed here except for $MIDMID$, only reposition the existing nodes.

In continuation with the above example, we have

$MID: 7 \rightarrow NIL$

Updated $LL1: 1 \rightarrow 4 \rightarrow 2 \rightarrow 8 \rightarrow 3 \rightarrow 7 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 10 \rightarrow 6 \rightarrow$

NIL $UpdatedLL1: 1 \rightarrow 4 \rightarrow 2 \rightarrow 8 \rightarrow 3 \rightarrow 7 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 10 \rightarrow 6 \rightarrow NIL$ $LL2:$

NIL $LL2: NIL$

$LL3: NIL$ $LL3: NIL$

###Input:

iv) Implement the following scenario.

There is a singly linked list ($LL1$)($LL1$) having $2 \times n$ nodes ($n \geq 1$).

Create two linked lists ($LL2$ and $LL3$)($LL2$ and $LL3$) each having $n - 1$ nodes. $LL2LL2$ and $LL3LL3$ are respectively formed by adding values of consecutive odd-positioned and even-positioned nodes in $LL1LL1$.

Note: Position of first node in $LL1$ is one.

- ☐ Line 1 contains an integer N . The total number of nodes in $LL1LL1$ is $2 \times N$.
- ☐ Line 2 contains $2 \times N$ integers separated by space. These are the node contents of $LL1LL1$ starting from head position.

###Output:

- ☐ Line 1 has $N - 1$ space separated integers, the contents of $LL2LL2$ starting from head position.
- ☐ Line 2 has $N - 1$ space separated integers, the contents of $LL3LL3$ starting from head position.
- ☐ Line 3 has an integer giving the value of $MIDMID$.
- ☐ Line 4 has space separated contents of $Updated LL1UpdatedLL1$ starting from head position.
- ☐ Line 5 has an integer giving total number of nodes created throughout the execution.

###Sample Input: 4 3 6 1 2 4 5 7 9

###Sample Output: 4 5 11 8 7 14 12 3 4 6 5 1 11 2 12 4 8 5 7 7 14 9 15

v) Transverse a matrix in spiral form and print the elements

2. Steps for experiment/practical/Code:

i)

```
#include <iostream> using  
namespace std;
```

```
int main() {  
    // your code goes  
here    int t;  cin>>t;  
    while(t--)  
    {  
        int n;  
cin>>n;    int x;  
int y;    int  
a[11]={0};    int  
sum=0;    for(int  
i=0;i<n;i++)  
    {  
cin>>x>>y;  
if(x<9&& a[x]<y)  
{    a[x]=y;  
    }  
    }  
    for(int i=0;i<9;i++)  
    {  
sum=sum+a[i];  
    }  
cout<<sum<<endl;
```

```
}  
return 0;}
```

ii)

```
#include <iostream>  
using namespace std;  
  
int main() {  
    // your code goes  
here    int t; cin>>t;  
    while(t--)  
    {  
        long long int n;  
cin>>n;  
        long long int a[n];  
        long long int s=0;  
for(int i=0;i<n;i++){  
cin>>a[i];    s+=a[i];  
        }  
for(int i=0;i<n;i++){  
    {  
        cout<<(s-a[i])<<" ";  
  
    }  
    cout<<endl;  
    }  
return 0;  
}
```

iii)

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    int n,i,j;    cin>>n;
    int arr[n][n];    int
    swaps=0;    int min;
    for(int m=0;m<n;m++)
    {
        for(int t=0;t<n;t++)
        {
            cin>>arr[m][t];
        }
    }
    int k=0;
    while(k<n-1)
    {
        for(int j=0;j<n;j++)
        {
            i=k+1;
            min=k;
            for(;i<n;i++)
            {
                if(arr[i][j]<arr[min][j])
                {
                    min=i;
                }
            }
            if(min!=k)
            {
```

```

        ++swaps;
swap(arr[k][j],arr[min][j]);
    }
}

for(int i=0;i<n;i++)
{
j=k+1;
min=k;
    for(;j<n;j++)
    {
        if(arr[i][j]<arr[i][min])
        {
min=j;
        }
    }
    if(min!=k)
    {
        ++swaps;
swap(arr[i][k],arr[i][min]);
    }

    ++k;

}
cout<<swaps<<endl;
for(int m=0;m<n;m++)
{
    for(int t=0;t<n;t++)
    {
        cout<<arr[m][t]<<" ";
    }
}
return 0;
}

```

iv)

```
#include <iostream> #include  
<bits/stdc++.h>  
using namespace std;
```

```
class node{  
public:  int data;  
node* next;  
node(int val){  
data = val;  
next = NULL;  
}  
};
```

```
void display(node* head){  
node* temp = head;  
while(temp!=NULL){  
cout<<temp->data<<" ";  
temp = temp->next;  
}  
cout<<endl;  
}
```

```
void insert(node* &head,int val){  
node* n = new node(val);  
if(head==NULL){ head = n;  
return;  
}  
node* temp = head; while(temp->  
next != NULL){  
temp=temp->next;  
}  
}
```



```

temp->next = n;
}

int length(node* head){
node* temp = head;
int count = 0;
while(temp!=NULL){
    count++;    temp=temp-
>next;
}    return
count;
}

int main() {
    int n; cin>>n;    node*
head = NULL;    for(int
i=0;i<(2*n);i++){        int
temp; cin>>temp;
        insert(head,temp);
    }

    node* head2 = NULL;    node*
head3 = NULL;    node* t1 =
head;    int count1 =0,sum1=0;
while(t1!=NULL){
count1++; sum1 += t1->data;
if(count1==2){
insert(head2,sum1);
    sum1=t1->data; count1=1;
    }
    t1 = t1->next->next;
}

    node* t2 = head->next;
count1 =0; sum1=0;
while(t2!=NULL){
count1++; sum1 += t2->data;
if(count1==2){
insert(head3,sum1);
    sum1=t2->data; count1=1;
    }
    if(t2->next == NULL) break;
}

```

```

    t2 = t2->next->next;
}

display(head2); display(head3);    node* mid = head;

while(mid->next != NULL){
    mid = mid->next;
}
cout<<(head->data + mid->data)<<endl;
node* newhead = NULL;    node* temp1
= head;    node* temp2 = head2;
while(temp2 != NULL){
insert(newhead,temp1->data);
insert(newhead,temp2->data);    temp1
= temp1->next;    temp2 = temp2-
>next;
}
insert(newhead,temp1->data);
insert(newhead,(head->data + mid->data));
temp1 = temp1->next;

temp2 = head3;    while(temp2
!= NULL){
insert(newhead,temp1->data);
insert(newhead,temp2->data);
temp1 = temp1->next;    temp2
= temp2->next;
}
insert(newhead,temp1->data);

display(newhead);
cout<<length(newhead)<<endl;

return 0;
}

```

v)

```
#include <bits/stdc++.h>
using namespace std;

int row, column;

int a[1000][1000];

void spiral(){
    int up=0, down=row-1, left=0, right=column-1;

    while(up<=down && left<=right){

        if(up<=down && left<=right){
            for(int j=left; j<=right; j++)
                cout << a[up][j] << " ";

            up++;
        }

        if(up<=down && left<=right){
            for(int i=up; i<=down; i++)
                cout << a[i][right] << " ";

            right--;
        }

        if(up<=down && left<=right){    for(int j=right;
j>=left; j--)
                cout << a[down][j] << " ";

            down--;
        }
    }
}
```

```

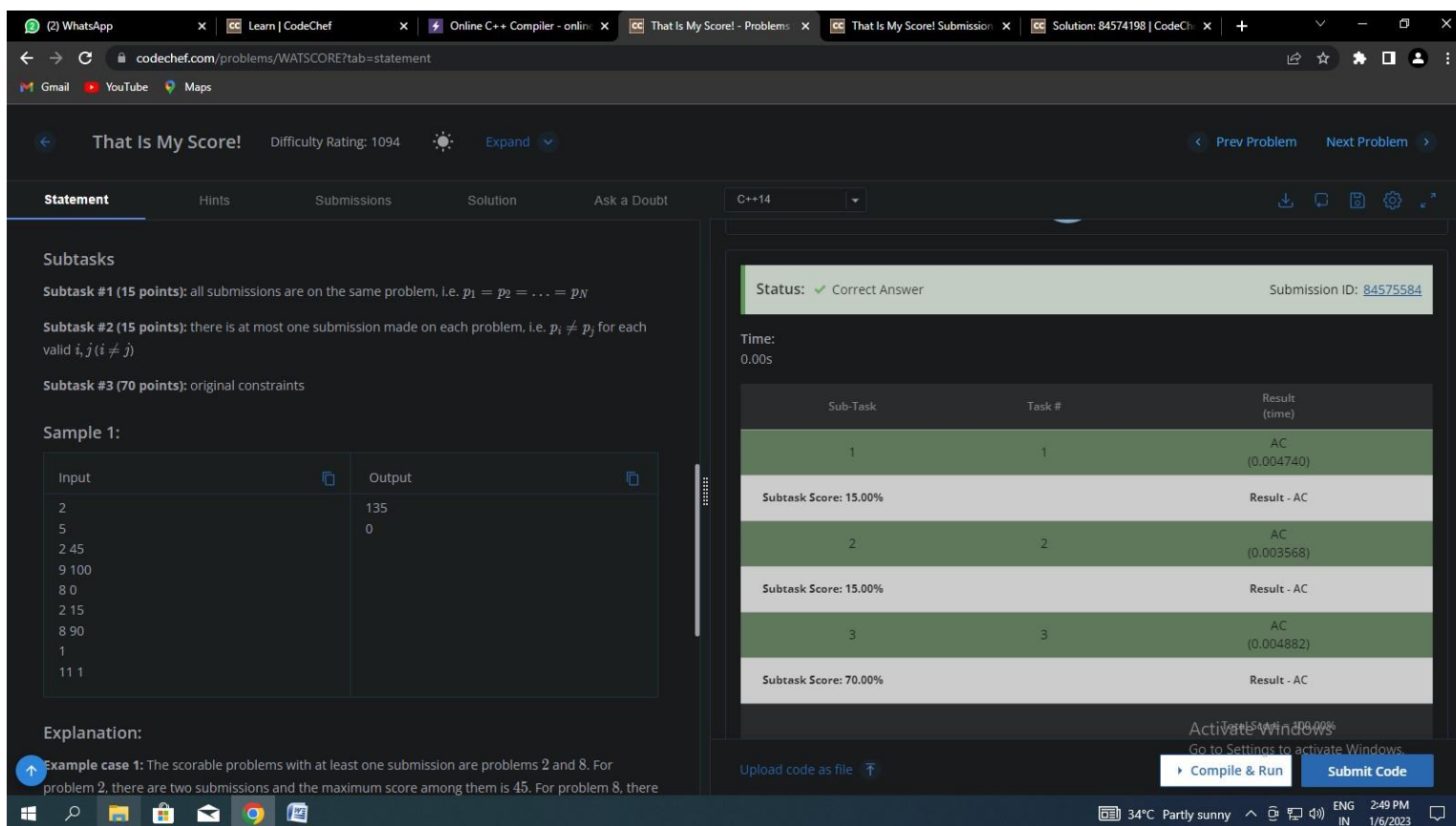
    }
    if(up<=down && left<=right){
    for(int i=down; i>=up; i--)
        cout << a[i][left] << " ";
        left++;
    }
}

int main() {
    int t;
    cin>>t;
    while(t--){
        cin >> row;
        column=row;
        for(int i=0; i<row; i++){
            for(int j=0; j<column; j++){
                cin>>a[i][j];
            }
        }
        spiral();
        cout<<"\n";
    }

    return 0;
}

```

3. Observations/Discussions/ Complexity Analysis: i)



The screenshot shows the CodeChef problem page for "That Is My Score!". The problem has a difficulty rating of 1094. The user has submitted a solution in C++14, which has been accepted (AC) with a submission ID of 84575584. The execution time is 0.00s.

Subtasks:

- Subtask #1 (15 points):** all submissions are on the same problem, i.e. $p_1 = p_2 = \dots = p_N$
- Subtask #2 (15 points):** there is at most one submission made on each problem, i.e. $p_i \neq p_j$ for each valid i, j ($i \neq j$)
- Subtask #3 (70 points):** original constraints

Sample 1:

Input	Output
2	135
5	0
2 45	
9 100	
8 0	
2 15	
8 90	
1	
11 1	

Explanation:

Example case 1: The scorable problems with at least one submission are problems 2 and 8. For problem 2, there are two submissions and the maximum score among them is 45. For problem 8, there

Subtask Results:

Sub-Task	Task #	Result (time)
1	1	AC (0.004740)
Subtask Score: 15.00%		Result - AC
2	2	AC (0.003568)
Subtask Score: 15.00%		Result - AC
3	3	AC (0.004882)
Subtask Score: 70.00%		Result - AC

Buttons: Compile & Run, Submit Code

ii)

The image is a screenshot of a web browser displaying the CodeChef 'The Array Problem' solution page. The browser's address bar shows the URL 'codechef.com/problems/ARRPROB?tab=statement'. The page has a dark theme. On the left, the 'Statement' tab is active, showing the problem description: Shahid is a computer science student who needs to solve a problem within a time limit. The problem statement asks for an array of positive integers A_1, A_2, \dots, A_N with length N , where each element is the sum of all other elements in the array. The input section specifies that the first line contains the number of test cases T , and each test case consists of two lines: the first line contains N (the length of the array) and the second line contains N space-separated positive integers. The output section states that for each test case, a single array of the same length should be printed. The constraints are $1 \leq T \leq 100$, $1 \leq N \leq 10^5$, and $0 \leq A[i] \leq 10^9$. On the right, the 'Submissions' tab is active, showing a successful submission with a 'Correct Answer' status, a submission ID of 84579792, and a time of 0.59s. A 'Problem Solver Badge' is also displayed, indicating that the user has solved 154 more problems to get a Silver Badge. The bottom of the page shows a Windows taskbar with various icons and a system tray displaying the date and time as 3:34 PM on 1/6/2023.

iv)

egov                   

codechef.com/problems/DSL1?tab=statement

Linking Linked List Difficulty Rating: NA Expand

Statement Submissions Solution C++14

Line 5 has an Integer giving total number of nodes created throughout the execution.

###Sample Input: 4 3 6 1 2 4 5 7 9

###Sample Output: 4 5 11 8 7 14 12 3 4 6 5 1 11 2 12 4 8 5 7 7 14 9 15

Did you like the problem statement?

Share **Pro** with your friend

- Add a friend while purchasing Pro
- Both you and your friend get **35% OFF**

Additional 20% OFF for coders with Bronze Badge

More Info

Contributors

Problem Solver Badge 98 / 250

Solve 152 more problems to get Silver Badge

Status: ✓ Correct Answer Submission ID: 84580385

Time: 0.00s

Sub-Task	Task #	Result (time)
1	0	AC (0.004066)
Subtask Score: 50.00%		Result - AC
2	1	AC (0.003970)

Upload code as file

Compile & Run Submit Code

34°C Partly sunny 3:41 PM 1/6/2023

v)

codechef.com/problems/SPMAT?tab=statement

Spiral Matrix Difficulty Rating: NA Expand

Statement Submissions Solution C++14

Problem

Transverse a matrix in spiral form and print the elements

Input

- The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follow.
- The first line of input contains an integer N denoting the number of rows/columns in the matrix.
- The next N lines contain the rows of the matrix. The lines contains space-separated integers denoting the elements in each row.

Output

For each test case output a single line of space-separated integers transversing the matrix in spiral form.

Constraints

- $1 \leq T \leq 20$
- $1 \leq N \leq 100$
- $1 \leq a_{ij} \leq 10^5$

Example

Test against Custom Input

```
3
1 2 3
4 5 6
7 8 9
```

Problem Solver Badge 99 / 250

Solve 151 more problems to get Silver Badge

Status: ✓ Correct Answer Submission ID: 84580623

Time: 0.00s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Activate Windows

Go to Settings to activate Windows

Upload code as file

Compile & Run Submit Code

34°C Partly sunny 3:45 PM 1/6/2023

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			