Debargest element in the array

int largest Element (vector cint> & avor, n) {

int largest = arr LoJ;

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

if (arr LiJ > largest)?

largest = arr LiJ;

riturn Sargest;

2 Second largest an smallest elements

vector < int > get3ec and Onder Elements (int n, vector < int > a) {

if ( n == 0 11 n == 1) }

cout <<-1 << endl;

a. sort ();
int ssmall = a [1];
int slargest = a [n-2].
eout << slargest << " "<< small << ind;

3 Check if away is sorted and notated

bool check (rectar < int>  $\times$  nums, n) {

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

if (nums IiI < nums Ii-1) {

neturn false;

}

return true;

1 Remove duplicates from sorted array

int removeDuplicates (vector < int > x avr, n) {
int i=0;

for (int j=1;  $j \times n$ ; j++)?

if (our I if I := aur I := a

} i++;

return i+1;

5 Left notate an array by one

an In-1] = temp;

netwin arr;

1 More zero's to end

vector 2 int> moveZoros (n, vector < int> a) {

int j = -1; for (int i = 0;  $i \ge n$ ; i++) } if (  $a \ge i \le -1$ ) } i = i.

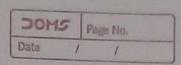
j=i; break;

for (int  $i = j+1 : i \le n : i++)$ }

if (aLiJ!=0')?

Swap (aLiJ, aLjJ). j++i j

return a;



Dinean Search

int search InBorted (int our II, int N, int X){

far (int i = 0; i × N; i++) {
 if (arr IiJ = = x) }
 return 1;
}

neturn -1;

1 3 ort an array of 0's 1's and 2's

void sortColars (rectar < int> 4 nums)  $\frac{1}{2}$  int  $C_1 = 0$ ,  $C_2 = 0$ ,  $C_3 = 0$ ;

int n = nums. size(); for (int i = 0; i < m; i + +)? if (nums IiJ = = 0)? C4 + + +

 $\frac{2}{3}$  else { nums YiJ = = 1}

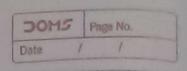
3 (2 ++ ;

for (int i = 0;  $( \times C1 ; i + + )$ ?

nums IiJ = 0;

for (int  $i = C1 ; i \times C1 + C2 ; i + + )$ ?

nums IiJ = 1;



```
for (int i = c1 + c2; i < n; i++)?

nums IiJ = 2;
```

3 Majarity element

int majarityElement (nector < int > v) {

map < int, int > mpp:

for (int i=0; i 2 v. size(); i++)?

mpp [v Ii]]++;

for (auto it: mpp) {
 if (it. second 7 (v. size() 12)) {
 return it. first;
 }

netwin -1;

10 hadane's algarithm: Maximum subcorrow

int maxSubarray (vector Lintz nums)?

long long maxi = LONG\_MIN;

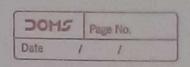
long long sum = 0;

for (int i= 0; i < mums. sigs (); i++)}

sum += nums Ii].

if (sum > masci) }

masci = sum.



if (sum 20)3

sam =0;

return maxi;

Deartrange the correy in alternating positive and negative items.

ours I pas I = nums I i ];

pas += 2;

neturn ans;

12 Leaders in an array

vector <int > Seaders (int n, int avr I J)?

vector < int > ans;

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

bool leader = true;

for (int s = i + 1; j = n; j + +) ?

if (arr Ij J > arr I i J)?

leader = false;

break;

}

if (leader)?

ans-push\_back (arr Ii J);

return ans;

}

B) Ratate motrix by 90 degrees

void rotate (vector < vector < int >> 3 matrice)?

int m = matrice. Size ();

far (int i = 0; i < n; i++)?

for (int j = 0; j \( \) i \( \) i

for (int i = 0; i < n; i++) }
reverse (matrix Ii]. begin, matrix Ii]. end ();

```
1 Next permutation
```

void mesetPormutation (vector < int >  $\frac{1}{2}$  nums)?

int m = nums. sige(); i = n-2;

whele  $(i > 0 = 0 = \frac{1}{2}$  nums  $TeJ > 0 = \frac{1}{2}$ int j = n-1;

while (nums TjJ < 0 = nums TiJ)?

Swap (nums Ii], numsIj]);

reverse (nums. begin () + i+1, nums. end()).

## (3) Count Subarrays with given sum.

int subarray 3 um (vector < int > 4 mums, int x)?

int  $n = \text{mums} \cdot \text{sigs}()$ ;

int cnt = 0;

for (int i = 0; i \( \) int \( \) sum = 0;

for (int j=i; j < n; j++){ Sum += mim s LjJ;

if (sum = = x){

cnt + +;

naturn ent;

## (1) Majority element (11/3 times)

exector Lint > majority Floment (vector Lint > v) {

int n = v. size()

vector Lint > b;

for (int i = 0, i < n, i++)?

if (b. size () = = 0 || ls IoJ! = vIiJ)?

int ent = 0;

for (int j = 0; j < n, j++)?

if (vIjJ = = vIiJ)?

? ent ++;

if (b. size () = = 2)?

break;

g retuin ls.

## (F) Merge overlapping Subintervals

vector < vector < int >> merge Over (vector < vector < int >> & arr )

int n = orr. size ();
Sort (arr. begin (), orr. end ()).
vector < vector < int >> ans:

(B) Finding missing and repeating numbers

vector < int > find Musing Repeating (vector < int > a) {

int n=a. size().

return ans;

int repeat = -1, missing = -1.

for (int i = 1, i <= n; i++) i <= 0;

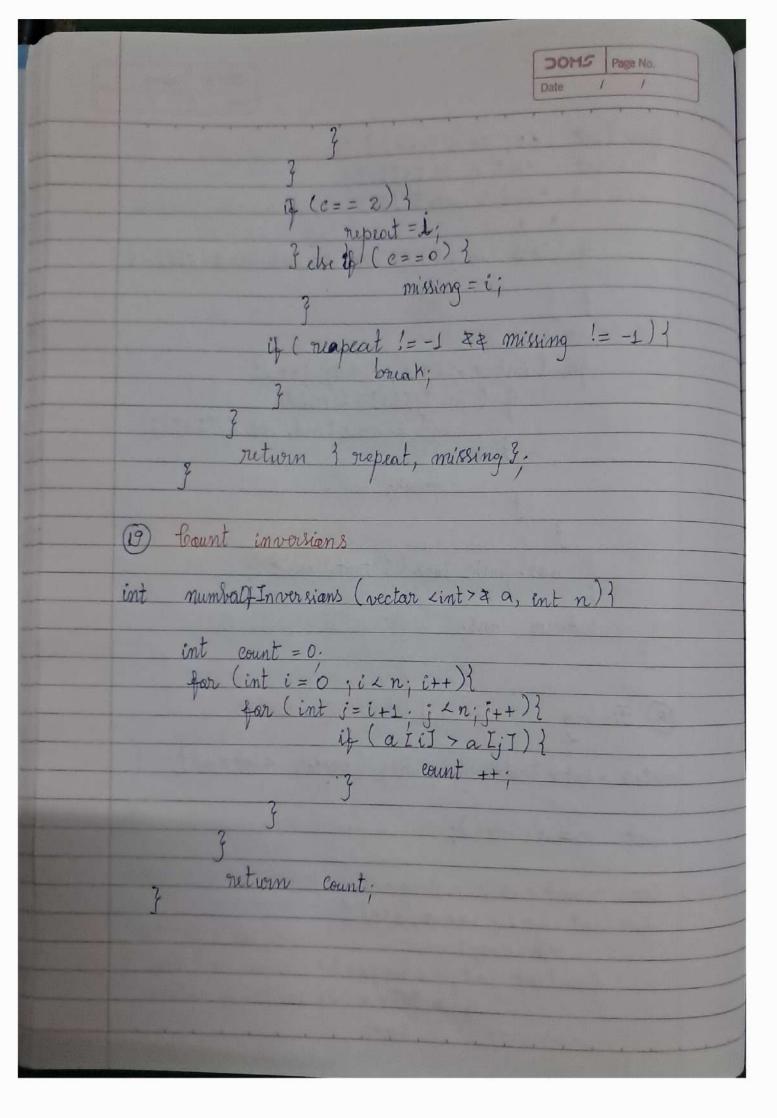
int c = 0;

far (int i = 0; i < n; i++) {
 int start = arr I i J I o I;

int end = arr I i J I 1 ];

continue;

for (int j=0; j < n; j++) {  $i \neq (a \ Ij \ J==i)$ }



(20) Revouse pairs

int toam (vector xint > = smill, int n) { int count = 0;

for (int i = 0; i < n; i++) { for (int j = i+1; j < n; j++)?

if (snill IiJ > e \* snill IjJ) { count ++;

return count;

21) Maseimum product Subarray

int subarray (vector < int > \$ arr)?

int result = arr LOJ;

for (int i=0; ix arr. sige()-1; i++) {

ent p = ar IiJ;

for (int j = i+1; j < avr. size(); j++)?result = max(result, p);

p \*= avr Ij J;

result = mase (result, p);

neturn result;