

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

3

JAI

BABA

KI

Main Ideas, Questions & Summary:

Library / Website Ref.:-

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* B2D \Rightarrow

$$\text{bit} = M \% 10;$$

$$\text{decimal} = \text{bit} * \text{pow}(2, i++) + \text{decimal};$$

$$M = M / 10;$$

* D2B \Rightarrow

$$\text{bit} = M \% 2;$$

$$\text{binary} = \text{bit} * \text{pow}(10, i++) + \text{binary};$$

$$M = M / 2;$$

OR

$$\text{bit} = \underbrace{(M \% 1)}_{\text{Internally stored as binary}};$$

$$\text{binary} = \text{bit} * \text{pow}(10, i++) + \text{binary};$$

$$M = M \gg 1; \quad (M = M / 2)$$

Main Ideas, Questions & Summary:

Library / Website Ref.: -

★ Odd / Even ⇒

if ($m \% 2$) == 0

Even;

else

Odd;

★ Sq Root for Prime No ⇒

for(i=2; (i <= sqrt(n)); i++) {}

<C math>

★ LeetCode 7 (Reverse No.)

★ Size of array in loop ⇒

for (i=0; i < (sizeof(arr)/4); i++) {}

★ Pattern Printing

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* XOR (Unique mess) \Rightarrow

$\Rightarrow \text{ans} = 0;$

$\text{for } \{ i=0; i < \text{arr.size()}; i++ \}$

$\text{ans} = \text{ans} \Delta \text{arr}[i];$

}

0	0	0
1	1	0
0	1	1
1	0	1

$$\Rightarrow x \wedge x = 0$$

$$x \wedge 0 = x$$

$$x \wedge y \Rightarrow x \rightarrow \begin{smallmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{smallmatrix}$$

$$(3) \quad (5) \quad y \rightarrow \begin{smallmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{smallmatrix} \quad (\text{XOR})$$

$$\frac{}{\begin{smallmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \end{smallmatrix}} \quad (6)$$

* Sets (Store Unique Element) \Rightarrow <Set> Header

\Rightarrow No duplicate allowed, (Don't support indexing $s[.]$)

\Rightarrow Element store in ascending order (Unordered-set)

\Rightarrow Can't modify element (only insert | remove)

\Rightarrow Set <int> s; $\Rightarrow s.\text{Count}(5) \Rightarrow s.\text{find}(40)$

\hookrightarrow ele present or not
 \hookrightarrow Return 0/1

\hookrightarrow Return address if found else point to s.end()

$\Rightarrow s.\text{insert}(5)$

$\Rightarrow s.\text{erase}(5)$

$\Rightarrow s.\text{clear}()$ Clear all data

$\Rightarrow s.\text{size}()$

$\Rightarrow s.\text{empty}()$

\hookrightarrow auto it = s.find(10)
if(it != s.end())
Found? else not

Main Ideas, Questions & Summary:

★ Map \Rightarrow <map> \rightarrow header (Same $f(n)$ as Set)

\Rightarrow Insertion, Deletion, Search $\rightarrow O(\log n)$

$\rightarrow O(1) \rightarrow$ Unordered-Map

\Rightarrow Map < int, string > mp;

mp[1] = "Apple";

mp[2] = "Orange";

\Rightarrow for (auto &p : mp) {

cout << p.first << " ->" << p.second;

}

map < int, int > mp;

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

mp[arr[i]]++;

\Rightarrow for (auto &p : mp) {

cout << mp[~~arr~~[i]];

★ Intersection of array / Vector \Rightarrow

↳ Using Set

↳ Using Map (unique intersection, As many intersection possible)

↳ Brute force (2 loops)

★ Union of array / Vector \Rightarrow

↳ Using Brute force

↳ Map

↳ Set

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ 2-D array \Rightarrow

↳ $(\text{Total column} * i) + j$ → Give index for 1D array
→ Use it to convert 2D to 1D

↳ arr[3][2] = {{1, 2}, {3, 4}, {5, 6}}

$$(2 * 1 + 1)$$

→ 3 index in 1D array

★ Transpose of array / Vector \Rightarrow

↳ vector <vector<int>> ans, col-size, vector<int>(row-size);
→ Attention!!
Use col-size
not row-size

↳ for(i=0; i < row-size; i++) {
 for(j=0; j < col-size; j++) {
 ans[j][i] = mat[i][j],
 }
}

Main Ideas, Questions & Summary:

★ Sort 0, 1, 2 (Dutch Flag algo) =>

↳ {0 1 2 0 1 2}

↳

```
Start = 0;
Mid = 0;
End = arr.size() - 1;
```

```
while (mid <= last) {
    if (arr[mid] == 0) {
        swap(mid, Start);
        Start++;
        mid++;
    }
}
```

```
else if (arr[mid] == 1) {
    mid++;
}
```

```
else {
    swap(mid, End);
    End--;
}
```

```
}
```

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* First Repeating Element \Rightarrow

↳ Brute Force (2 loop)

↳ Map

* Common Element in 3 Sorted array \Rightarrow

↳ Map (Uses extra space)

↳ 3 Pointer

↳ $i = j = k = 0;$

while ($i < arr_1$ $\&&$ $j < arr_2$ $\&&$ $k < arr_3$) {

 if ($arr_1[i] == arr_2[j]$ $\&&$ $arr_2[j] == arr_3[k]$) {

 ans.push(arr_1[i]);

 i, j, k++;

}

 else if ($arr_1[i] < arr_2[j]$) {

 i++;

}

 else if ($arr_2[j] < arr_3[k]$) {

 j++;

}

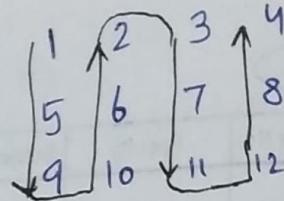
 else {

 k++;

 }

Main Ideas, Questions & Summary:

* Wave Print matrix =>



1 5 9 10 6 2 3 7 11 12 8 4

↳ void print (vector<vector<int>> arr) {

for (col = 0; col < arr[0].size(); col++) {

if ((col % 2) == 0) { // Even Col

for (row = 0; row < arr.size(); row++) {

cout << arr[row][col];

}

↓
Don't interchange

else {

for (int row = arr.size() - 1; row >= 0; row--) {

cout << arr[row][col];

}

}

}

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Spiral Matrix \Rightarrow

\hookrightarrow vector<int> print (vector<vector<int>> arr) {

vector<int> ans;

int row_size = arr.size();

int col_size = arr[0].size();

int start_row = 0;

int end_row = row_size - 1;

int start_col = 0;

int end_col = col_size - 1;

int total_element = row_size * col_size;

int count = 0;

while (total_element > count) {

for (int i = start_col; i <= end_col && total_element > count; i++) {

ans.push_back(arr[s_row][i]);

(count++);

} (start_row++);

for (int i = s_row; i <= e_row; i++) {

ans.push_back(arr[i][e_col]);

(count++);

} (e_col--);

for (int i = e_col; i >= s_col; i--) {

ans.push_back(arr[e_row][i]);

(count++);

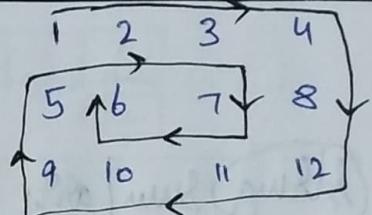
} (e_row--);

Main Ideas, Questions & Summary: for (int i = e_row; i >= s_row; i--) {

ans.push_back(arr[i][start_col]);

Count++;

} (start_col++);



* Add 2 no. represented by 2 array \Rightarrow

0	9	0	0	3	5
---	---	---	---	---	---

2	2	7		
9	0	2	6	2

+

Ans

↳ String sum(arr1, arr2) {

String ans;

 int i = arr1.size() - 1;

 int j = arr2.size() - 1;

int carry = 0;

while(i >= 0 || j >= 0) {

 int n = a[i] + b[j] + carry;
 i--, j--;

 int digit = n % 10;

 ans.push_back(digit + '0');

 carry = n / 10;

}

while(i >= 0) {

}

while(j >= 0) {

}

if(carry) {

 ans.push_back(carry + '0');

}

while(ans[ans.size() - 1] == '0') {

 ans.pop_back();

reverse(ans.begin(), ans.end());

return ans;

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Factorial of large no. \Rightarrow

$$\begin{array}{|c|c|c|} \hline x & y & z \\ \hline \times & a & \\ \hline \end{array}$$

nyza

\hookrightarrow vector<int> ans(int n){

vector<int> ans{1};

int carry = 0;

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

 for (j=0; j < ans.size(); j++) {

 int digit = ans[i] * i + carry;

 ans[j] = digit % 10;

 carry = digit / 10;

}

 while (carry) {

 ans.push_back(carry % 10);

 carry /= 10;

}

 reverse(ans.begin(), ans.end());

 return ans;

}

Main Ideas, Questions & Summary:

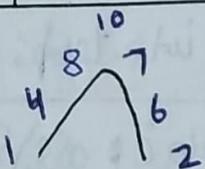
★ First & Last in sorted array \Rightarrow

↳ int search (vector<int> arr, target, bool left - search) {
 int start = 0;
 int end = arr.size() - 1;
 int ans = -1;
 while (start <= end) {
 int mid = start + (end - start) / 2;
 if (mid > target)
 end = mid - 1;
 else if (mid < target)
 start = mid + 1;
 else {
 ans = mid;
 if (left - search)
 end = mid - 1;
 else
 start = mid + 1;
 }
 }
 return ans;
}

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Peak element in array \Rightarrow



int find (arr){

start = 0;

end = arr.size() - 1;

while (start < end){

if (arr[mid] < arr[mid + 1]) {

start = mid + 1;

}

else {

end = mid;

}

}

return start;

}

Main Ideas, Questions & Summary:

Library / Website Ref.:-

★ Move all -ve to end (Preserve order) =>

↳ void (arr){

vector<int> temp;

for (auto i : arr){

if (i >= 0)

}

temp.push_back(i);

for (auto i : arr){

if (i < 0)

temp.push_back(i);

}

for (i = 0; i < arr.size(); i++) {

arr[i] = temp[i];

}

}

★ Move all -ve to end (Order not preserve) =>

↳ int n = arr.size();

int i = 0, j = 0;

while (j < n){

if (arr[j] >= 0){

if (i != j)

swap(i, j);

i++;

}

j++;

↳ int i = 0;

j = 1;

while (j < arr.size()) {

if (arr[j] < 0)

j++;

else {

swap(arr[i], arr[j]);

i++;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Pivot element in rotated & sorted array \Rightarrow

\hookrightarrow int pivot(arr){

start = 0;

end = arr.size() - 1;

while (start <= end) {

if (start == end)

return start;

IMP!!!

Pivot

5 6 7 1 2 3 4

Use these condition
just after if(i) otherwise
error in LC (Unbounded)

if (arr[mid] > arr[mid + 1] && (mid + 1 <= end)) {

return mid;

}

if (arr[mid] < arr[mid - 1] && (mid - 1 >= start)) {

return mid - 1;

}

if (arr[start] > arr[mid])

end = mid - 1;

else

start = mid + 1;

}

return -1;

* Search in Rotated & Sorted array

Main Ideas, Questions & Summary:

★ SqRt using BS \Rightarrow

↳ int sqrt(m){

```
start = 1; end = m;  
int ans = -1;  
while (start <= end) {  
    if ((mid * mid) == m)  
        return mid;  
    else if ((mid * mid) > m) {  
        end = mid - 1;  
    } else {  
        ans = mid;  
        start = mid + 1;  
    }  
}  
return ans;
```

↳ int main(){

```
int number = 10;  
int result = sqrt(number);  
int precision = 5;  
float final = result;  
float step = 0.1;
```

```
for (int i = 0; i < precision; i++)
```

```
for (float j = final; j * j <= number; j += step) {  
    final = j;
```

```
step /= 10;
```

```
}
```

```
cout << final;
```

```
for (int i = 0; i < precision; i++)  
    while (true)
```

```
if (final * final <= number)
```

```
    final += step;
```

```
else
```

```
    final -= step;
```

```
}  
break;
```

```
Step /= 10;
```

```
}
```

```
cout << final;
```

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ 2) binary Search =

↳ Bool Search (arr [] [u], row, col, target) {

start = 0;

end = row * col - 1;

mid = start + (end - start) / 2;

while (start <= end) {

int rowIndex = mid / col;

int colIndex = mid % col;

if (target == arr [rowIndex] [colIndex]);

return true;

else if (target > arr [rowIndex] [colIndex]) {

start = mid + 1;

else

end = mid - 1;

mid = start + (end - start) / 2;

}

return false;

}

Main Ideas, Questions & Summary:

* check whether a no. can be expressed as sum of 2 prime no. or non-prime no. \Rightarrow

↳ IsPrime(m){

if ($m \geq 1$)

return False;

for ($i = 2$; $i \leq \sqrt{m}$; $i++$){

if ($m \cdot i \cdot i == 0$)

return True;

return False;

↳ CanExpress(m){

for ($i = 1$; $i < m$; $i++$){

$j = m - i$;

if (((isprime(i) && isprime(j)) || (!isprime(i) && !isprime(j))) {

cout << i << " " << j;

return "Yes";

}

return "No";

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Binary Search in a nearly sorted array \Rightarrow

{ 10, 3, 40, 20, 50, 80, 70 }

\hookrightarrow int search (arr, target) {

 l = 0, r = arr.size() - 1, mid;

 while (l < r) {

 if (target == arr[mid])

 return mid;

 else if (mid - 1 >= l && target == arr[mid - 1])

 return mid - 1;

 else if (mid + 1 <= r && target == arr[mid + 1])

 return mid + 1;

 else if (target < arr[mid])

 end = mid - 2;

 else

 start = mid + 2;

}

 return -1;

}

Main Ideas, Questions & Summary:

★ Divide 2 no. using BS \Rightarrow # LC = 29 # Check note in question.
↳ condition

↳ int divide(dividend, divisor){

long long absDividend = abs((long long)dividend);
long long absDivisor = abs((long long)divisor);

long long s = 0,
" " l = absDividend,
" " mid = s + (l - s)/2,
" " ans = 0;

Divisor $\frac{\text{Dividend}}{\text{Quotient}}$
Reminder

while(s <= l){

if (mid * absDivisor == absDividend)

ans = mid;

break;

else if (mid * absDivisor < absDividend)

ans = mid;

s = mid + 1;

else

e = mid - 1;

}

if ((dividend > 0 & divisor > 0) || (dividend < 0 & divisor < 0)){

if (ans > INT_MAX) (INT_MIN, -1)

return INT_MAX;

INT_MAX

return ans;

3

else

return -ans;

3

Precision lost

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Single element in sorted array $\Rightarrow \#LC \Rightarrow \underline{\underline{540}}$

```

int find (vector<int> nums) {
    l = 0;
    e = nums.size() - 1;
    while (l <= e) {
        if (l == e)
            return nums[l];
        mid = l + (e - l) / 2;

        if ((mid & 1) == 0) {
            if (nums[mid] == nums[mid + 1])
                l = mid + 2;
            else
                e = mid;
        }
        else {
            if (nums[mid] == nums[mid - 1])
                start = mid + 1;
            else
                e = mid - 1;
        }
    }
}

```

Main Ideas, Questions & Summary:

★ K-diff pairs in array \Rightarrow # LC \Rightarrow 532

↳ 2 pointer approach \Rightarrow

```
int pairs (vector<int> &nums, int target){
```

```
    sort (nums.begin(), nums.end());
```

```
    set<pair<int, int>> ans;
```

```
    int i = 0;
```

```
    int j = 1;
```

```
    while (j < nums.size()) {
```

```
        if (nums[j] - nums[i] == target) {
```

```
            ans.insert({nums[i], nums[j]});
```

```
        }  
        i++; j++;
```

```
    } else if (nums[j] - nums[i] > target)
```

```
        i++;
```

```
    else
```

```
        j++;
```

```
        if (i == j)  
            j++;
```

```
}
```

```
return ans.size();
```

```
}
```

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

↳ Binary Search ↳

```
int search (vector<int> &arr, int target) {
```

l = arr.size() - 1;

```
while (l >= r) {
```

mid = l + (l - r) / 2;

if (arr[mid] == target)

return mid;

else if (arr[mid] < target)

start = mid + 1;

else

end = mid - 1;

}

return -1;

}

```
int findpair (vector<int> &arr, int k) {
```

sort (arr);

set<pair<int, int>> ans;

```
for (int i = 0; i < arr.size(); i++) {
```

if (search (arr, i + 1, arr[i] + k) != -1)

ans.insert ({arr[i], arr[i] + k});

}

return ans.size();

Main Ideas, Questions & Summary:

★ Find k closest element \Rightarrow # LC \Rightarrow 658

↳ Brute force \Rightarrow

vector < int > find (all, int k, n) {

vector < pair < int, int >> order;

for (i = 0; i < all.size(); i++) {

} order.push_back ({all(n - all[i]), i});

sort (order.begin(), order.end());

for (auto i : order)

cout << i.first << " " << i.second << endl;

vector < int > ans;

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

ans.push_back (all[order[i].second]);

sort (ans.begin(), ans.end());

return ans;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

↳ 2 pointer \Rightarrow

vector <int> find (arr, k, n){

s = 0;

e = arr.size() - 1;

while ((e - s + 1) > k){

(if ($n - arr[s] > arr[e] - n$)

s++;

else

e--;

}

return vector <int> (arr.begin() + s, arr.begin() + end + 1);

OR

vector <int> ans;

while (s <= e)

ans.push_back (arr[s++]);

return ans;

Main Ideas, Questions & Summary:

Library / Website Ref.: -

★ Exponential Search \Rightarrow Better than BS if arr size is large & target at start

\Rightarrow Used in infinite (unbounded) array.

\hookrightarrow int binarySearch (arr, target, l, r){
 \dots
}

\hookrightarrow int exponentialSearch (arr, target) {

 if (arr[0] == target)

 return 0;

 int arrSize = arr.size() - 1;

 int i = 1;

 while (i < arrSize && arr[i] < target) {

 i *= 2; || i << 2;

}

 return binarySearch (arr, target, i / 2, min (i, arrSize));

}

if 'i' goes greater than arr size then
arrSize will be considered

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Book Allocation $\Rightarrow \#LC \Rightarrow 410$

V.V. Important

→ Book is Possible Sol (numb, K, mid) {

int pageSum = 0;

int student = 1;

for (i=0; i < numb.size(); i++) {

if (numb[i] > mid)

return false;

if (pageSum + numb[i] > mid) {

student++;

pageSum = numb[i];

if (student > k)

return false;

}

else {

pageSum += numb[i]; }

}

return true;

}

Pro →

Main Ideas, Questions & Summary:

Library / Website Ref.:-

↳ int bookAllocate (muns, k) {

if ($k > \text{muns.size}()$)
return -1;

→ Point use $\text{muns.size}() - 1$ as k start
from 'not '0'.

int ans = -1;

int s = 0;

int e = accumulate (muns.begin(), muns.end(), 0);

→ Add all
elements of
muns.

while ($s \leq e$) {

mid = start + $(e-s)/2$;

if (ispossibleSol (muns, k, mid)) {

ans = mid;

e = mid - 1;

}

else {

start = mid + 1;

}

}

return ans;

}

* Painters Partition Problem \Rightarrow Same as Book Allocation

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Char Array \Rightarrow

char name[100];

name[0] = 'a';

name[1] = 'l';

cin \gg name; // if space is used, it will be considered as null character
 '\\0'

cout \ll stlens(name); // tell how much char name is filled (size)

stlens(name), sizeof(name) are different.

* Char I/P with Spaces \Rightarrow

char name[100];

cin.getline(name, 10); // take 10 space only in char name

* cin \gg s; \rightarrow Works for single word (stops at space)

* getline(cin, s) \rightarrow Works for full line (include spaces)

Main Ideas, Questions & Summary:

Library / Website Ref.: -

★ No. of element present in char array \Rightarrow

★ Built-in fn \Rightarrow

$\text{cout} \ll \text{strlen}(\text{arr});$

★ Function \Rightarrow

$\hookrightarrow \text{int noElement(char arr[])}\{$

$\text{for } i=0; \text{ arr}[i] != '\backslash 0'; i++)\{$
 element++;

}

 return element;

OR

$i=0; \text{ element}=0;$
 $\text{while } (\text{arr}[i] != '\backslash 0')\{$
 element++;

}

 return element;

}

int main()\{

 cout \ll noElement(arr);

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Reverse String =>

↳ void ReverseString(char arr[]){

i=0;

j= strlen(arr)-1;

while(i < j){

swap(arr[i], arr[j]);

i++;

j--;

}

}

* Replace Spaces =>

↳ void Replace(char arr[]){

i=0;

while (arr[i] != '\0') {

if (arr[i] == " ")

arr[i] = '@';

i++;

}

}

Main Ideas, Questions & Summary:

Library / Website Ref.: -

★ Palindrome =>

↳ void palindrome (char arr[]) {

i = 0

j = strlen(arr) - 1;

while (i < j) {

if (arr[i] != arr[j])

return 0;

} i++, j--;

return 1;

}

★ Convert to Upper Case & Lower Case =>

↳ void convertUpper (char arr[]) {

for (i = 0; i < strlen(arr); i++) {

if (arr[i] != ' ') {

arr[i] = arr[i] - ('a' - 'A');

}

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-
<p>* <u>Difference b/w String, length, sizeof</u> \Rightarrow</p> <p style="text-align: center;">↓ ↓ ↓</p> <p style="text-align: center;">char[] string Both</p> <p>↳ char name[] = "Mayur Soni";</p> <p>cout << strlen(name); // 10 (count characters till '\0')</p> <p>* cout << sizeof(name); // 11 (total memory) (10 + '\0')</p> <p>↳ string name = "Mayur Soni";</p> <p>cout << name.length(); // 10 (no. of characters)</p> <p>cout << sizeof(name); // 32 (Size of string object)</p> <p>* <u>Remove all adjacent Duplicates in string \Rightarrow #LC \Rightarrow 1047</u></p> <p>↳ string remove(string s){</p> <p>string ans = "";</p> <div style="border: 1px solid black; padding: 10px;"> <pre>for(i=0; i < s.length(); i++){ if(ans.length() > 0 && ans[ans.length()-1] == s[i]) ans.pop_back(); else ans.push_back(); }</pre> <p style="text-align: right;">} * Can done with stack also.</p> </div> <p>Main Ideas, Questions & Summary:</p> <p>Library / Website Ref.:-</p>						

* Important f(n) of string =>

↳ substr =>

↳ string name = "Mayur Soni";

cout << "First" << name.substr(0,5); // Mayur

cout << "Second" << name.substr(6); // Soni

↳ compare =>

↳ 0 for equal

<0 if s1 is smaller and unmatched

>0 if s1 is larger & unmatched

↳ s1 = "aaacd";

s2 = "abcd";

s1.compare(s2); // -1

s1.compare(2,2,s2); // 1 (compare cd with abcd)

↳ s1 = "abcdefghijklmnopqrstuvwxyz";

s2 = "abcd";

s1.compare(s2); // 8

s1.compare(2,2,s2); // 1

↳ npos => not found

=> till the end

=> use as "string::npos" always

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

↳ push-back | Pop-back =>

↳ $s1 = "Mayur";$

$s1; // Mayur$

$s1.push_back('S');$ // Add 1 char at a time, append or '+=' is use for adding a string

$s1; // MayurS$

$s1.pop_back();$

$s1; // Mayur$

↳ find =>

↳ $s1 = "Hello my name is Mayur Soni";$

$int pos = s1.find("lo");$

↳ $(pos != \text{string}::npos)$ → Always use it

$\text{cout} \ll "Found at index" \ll pos \ll endl; // 3$

else

$\text{cout} \ll "Not found";$

↳ $s1.find('m', 9); // 11$

Main Ideas, Questions & Summary:

↳ Replace \Rightarrow

↳ $s = "I love Java";$

$s.replace(' ', '+');$

$cout \ll s; // I love +$

↳ erase \Rightarrow

↳ $s = "Hello World";$

$s.erase(5);$

$s; // He$

$s.erase(2, 2);$

$s; // He$

* Remove all occurrence of substring \Rightarrow #LC 1910

↳ String remove (String s, String part) {

int pos = s.find(part);

while (pos != string::npos) {

s.erase(pos, part.length());

pos = s.find(part);

}

return s;

}

Done by Recursion

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Valid Palindrome II $\Rightarrow \# LC = 680$

↳ Bool checkpalindrome (s, start, end) {

 while (start < end) {

 if (s[start] == s[end])

 start++, end--;

 else

 return false;

 }

 return true;

}

↳ Bool validity (string s) {

 start = 0;

 end = s.length() - 1;

 while (start < end) {

 if (s[start] == s[end])

 start++, end--;

 else

 return checkpalin(s, start+1, end) || checkpalin(s, start, end-1);

 }

 return true;

3

Main Ideas, Questions & Summary:

★ stoi \Rightarrow string to int [stoi(x)]

★ to_string \Rightarrow int to string [to_string(x)]

★ Minimum Time Difference \Rightarrow # LC \Rightarrow 539

↳ int find (vector<string>& timePoints) {

 vector<int> minutePoints;

STEP-I

 for (i=0; i < timePoints.size(); i++) {

 hourPoint = stoi (timePoints[i].substr(0, 2));

 minutePoint = stoi (timePoints[i].substr(3, 2));

 minutePoints.push_back(hourPoint * 60 + minutePoint);

}

STEP-II

 sort(minutePoints);

STEP-III

 int minimumMinute = INT_MAX;

 for (i=0; i < minutePoints.size() - 1; i++) {

 minimumMinute = min (minimumMinute, minutePoints[i+1] - minutePoints[i]);

}

STEP-IV

 int firstLastDiff = (minutePoints[0] + 1440) - (minutePoints[minutePoints.size() - 1]);

 minimumMinute = min (minimumMinute, firstLastDiff);

 return minimumMinute;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Print all substrings \Rightarrow

\hookrightarrow vector < string > Substring (String name) {

vector < string > subText;

for (i=0 ; i < name.length(); i++) {

 for (j = i+1; j <= name.length(); j++) {

 subText.push_back(name.substr(i, j-i));

}

return subText;

}

int main() {

 name = "abcd";

 vector < string > name2 = Substring(name);

 for (auto i : name2) {

 cout << i;

}

}

Main Ideas, Questions & Summary:

★ Palindromic Substring \Rightarrow # LC \Rightarrow 647

★ By creating substring \Rightarrow

↳ Is s Palindrome(s) {

}

↳ int countSubString (s) {

 Count = 0;

 for (i=0; i < s.length(); i++) {

 for (j=i; j < s.length(); j++) {

 if (isPalindrome (s.substring(i, j-i+1)))

 Count ++;

}

 return Count;

}

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Expand Around Index Approach $\Rightarrow O(n^2)$

↳ int expandAroundIndex(s, left, right){

Count = 0;

while(left >= 0 && right < s.length() && [s[left] == s[right]]){

Count++;

left--, right++;

}

return count;

}

↳ int countString(s){

totalCount = 0;

for(i=0; i < s.length(); i++){

int oddAns = expandAround([s, i, i]);

totalCount += oddAns;

int evenAns = expandAround([s, i, i+1]);

totalCount += evenAns;

}

return totalCount;

}

Main Ideas, Questions & Summary:

★ Custom Comparator \Rightarrow

\hookrightarrow `bool desc (int first, int second) {
 return first > second;
}`

\hookrightarrow `int main () {`

`vector<int> arr;`

`sort (arr.begin(), arr.end()); // Asc sort`

`sort (arr.begin(), arr.end(), desc); // Desc sort`

`}`

★ Valid Anagram \Rightarrow # LC \Rightarrow 242

\hookrightarrow `bool Anagram (string s, string t) {`

`sort(s), sort(t);`

`if (s == t) return true;`

`return false;`

`}`

\hookrightarrow `bool Anagram (s, t) {`

`unordered_map<char, int> check;`

`for (auto i : s) check[i]++;`

`for (auto i : t) check[i]--;`

`for (auto i : check) {`

`if (i.second != 0)`

`return false;`

`}` return true;

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Reverse Only Letters \Rightarrow # LC \Rightarrow 917

↳ String Reverse Letter (s) {

start = 0, end = s.length() - 1;

while (start < end) {

if (isalpha(s[start]) && isalpha(s[end])) {

swap(s[start], s[end]);

start++, end--;

}

else if (!isalpha(s[start]))

start++;

else

end--;

}

return s;

}

Main Ideas, Questions & Summary:

* Reverse vowel of string $\Rightarrow \#LC \Rightarrow \underline{345}$

↳ Local isVowel (char ch) {

ch = !isVowel (ch);

Swap b(n)

return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';

}

↳ String reverse (String s) {

start = 0, end = s.length() - 1;

while (start < end) {

if (!isVowel(s[start]) && isVowel(s[end])) {

Swap (s[start], s[end]);

start++, end--;

}

else if (!isVowel(s[start]))

start++;

else

end--;

}

return s;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Longest Common Prefix $\#LC \Rightarrow 14$

↳ 1st Approach :-

↳ String longest (vector<String> & strs) {

 String ans;

 int i = 0;

 while (true) {

 char current_char = 0;

 for (auto word : strs) {

 if (i >= word.length())

 return ans;

 else if (current_char == 0)

 current_char = word[i];

 else if (word[i] != current_char)

 return ans;

}

 ans.push_back (current_char);

 current_char = 0;

 i++;

}

 return ans;

}

Main Ideas, Questions & Summary:

↳ 2nd Approach →

↳ string longest (vector < string > & stbs) {

 string ans;

 int i = 0;

 while (true) {

 char current_char = stbs[0][i];

 for (auto word : stbs) {

 if (i >= word.length())

 return ans;

 else if (word[i] != current_char)

 return ans;

 }

 ans.push_back(current_char);

 i++;

 }

 return ans;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Isomorphic Strings \Rightarrow # LC \Rightarrow 205

↳ 1st Approach:

↳ bool isIsomorphic (String s, String t) {

 int hash[256] = {0};

 bool isTCharMapped[256] = {0};

 for (i=0; i < s.length(); i++) {

 if (hash[s[i]] == 0 && isTCharMapped[t[i]] == 0) {

 hash[s[i]] = t[i];

 isTCharMapped[t[i]] = true;

 }

 for (i=0; i < s.length(); i++) {

 if (hash[s[i]] != int(t[i]))

 return false;

 }

 return true;

}

Main Ideas, Questions & Summary:

↳ 2nd approach (Hash map)

↳ bool isIsomorphic (string s, string t) {

 unordered_map<char, char> map;

 unordered_map<char, bool> isTCharMapped;

 for (i=0; i < s.length(); i++) {

 if (map[s[i]] == map.end()) s is TCharMapped.

 map[s[i]] = t[i];
 isTCharMapped[t[i]] = true;

 }

 for (i=0; i < s.length(); i++) {

 if (map[s[i]] != (t[i]))

 return false;

 }

 return true;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Longest Palindromic Substring $\Rightarrow \# LC \Rightarrow 5$

\hookrightarrow Isol isPalindrome (String s, int start, int end) {

}

\hookrightarrow String longestPalindrome (String s) {

String ans = "";

for (i=0; i < s.length(); i++) {

 for (j=i; j < s.length(); j++) {

 if (isPalindrome (s, i, j)) {

 String t = s.substring (i, j-i+1);

 ans = t.length() > ans.length() ? t : ans;

}

}

 return ans;

}

Main Ideas, Questions & Summary:

Library / Website Ref.: -

★ Index of first occurrence in string $\Rightarrow \#LC \Rightarrow 28$

★ 1st approach \Rightarrow

↳ int strstr(string h, string m){
 if(h.length() < m.length())
 return -1;

 for(i=0; i <= h.length() > m.length(); i++){

 for(j=0; j < m.length(); j++){

 if(m[j] != h[i+j])

 break;

 if(j == m.length() - 1)

 return i;

}

 return -1;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ 2nd Approach \Rightarrow

\hookrightarrow int substr (string h, m) {

if (h.length() < m.length())

return -1;

for (i=0; (i <= h.length() - m.length()); i++) {

if (h.substr (i, m.length()) == m)

return i;

}

return -1;

}

★ 3rd approach \Rightarrow

\hookrightarrow int substr (h, m) {

int position = h.find (m);

return (position != string::npos) ? position : -1;

}

Main Ideas, Questions & Summary:

Library / Website Ref.:-

★ Reorganize String $\Rightarrow \#LC \Rightarrow 767$

↳ String reorganizeString (string s) {

// Count appearance of characters.

int hash[26] = {0};

for (auto i : s)

hash[i - 'a']++;

// find most frequent char

char max_freq_char;

int max_freq = INT_MIN;

for (i = 0; i < size(hash) | size(int); i++) {

if (max_freq < hash[i]) {

max_freq_char = i + 'a';

max_freq = hash[i];

}

}

// fill alternates with most freq char

int index = 0;

while (max_freq && index < s.length()) {

s[index] = max_freq_char;

index += 2;

max_freq--;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

// if most freq char not filled all then return "" else makes its freq to '0' .

if (max - freq != 0)

return " ";

else

hash [max - freq - char - 'a'] = 0 ;

// Now, fill remaining alternate

for (i=0 ; i < sizeof (hash) / sizeof (int) ; i++) {

while (hash [i] > 0) {

* (index = index >= s.length() ? 1 : index);

s [index] = i + 'a' ;

hash [i] -- ;

index += 2 ;

}

}

return s ;

}

Main Ideas, Questions & Summary:

★ Integer to Roman $\Rightarrow \# LC \Rightarrow 12$

↳ String toRoman(num){

String roman[] = {M, CM, D, CD, C, XC, L, XL, X, IX, V, IV, I};

int values[] = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};

Learn it

String ans = "";

for(i=0; i < size(roman) / size(values); i++) {
 while(num >= values[i]) {

ans += roman[i];

num -= values[i];

}

}

return ans;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Largest Number $\Rightarrow \# LC \Rightarrow 179$

↳ String sorting is different (Demographical sorting)

↳ Static void custom_Sort (String Sa, String Sb) {

 String S1 = a+b;

 String S2 = b+a;

 return S1 > S2; // if true, return 'a' else 'b'

}

↳ String largest (vector<int> Snum) {

 vector<string> Snumstr;

 for (auto Si : num)

 Snumstr.push_back(to_string(i));

 sort (Snumstr.begin(), Snumstr.end(), custom_Sort);

 // if first element after desc sort is '0' means
 // whole vector is of '0'.

 if (Snumstr[0] == "0")

 return "0";

 String ans = "";

 for (auto i : Snumstr)

 ans += i;

 return ans;

33

Main Ideas, Questions & Summary:

ppr & s. H & Dabach -
(white bridge and) bridge at pellat pellat
{ (H pellat at pellat) tch. mukur. Tch. site
tch. mukur. tch. mukur. tch. mukur.
tch. mukur. tch. mukur.
'o' site 'o' mukur. and q. 'o' site 'o' mukur.

{ (long & low sites) Dabach pellat
tch. mukur. tch. mukur. tch. mukur.
tch. mukur. tch. mukur. tch. mukur.
(H) pellat at tch. site tch. mukur.
{ (tch. mukur. (H) site tch. mukur. Dabach, Tch. site) tch.
mukur. 'o' at tch. site tch. mukur. tch. site q. 'o'
'o' tch. site tch. mukur.
("o" - (o) mukur) - q.
("o" mukur)
"o" - site pellat
(tch. site : i site) q.
i - site
site mukur.
quadruped & omnivorous.

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Pointers ⇒

- ↳ Size of any type of pointer (including array pointer) is always 8 because it stores address & address is of 8 bytes.
- ↳ Size also depends upon architecture.
- ↳ $\text{int } a = 5;$
 $\text{cout} \ll a; \quad // 5$
 $\text{cout} \ll \&a; \quad // \text{address of } a (\text{on } 61 \& 23)$
- ↳ $\text{int } *x = \&a;$
 $\text{cout} \ll x; \quad // (\text{on } 61 \& 23) \rightarrow \text{address of } a$
 $\text{cout} \ll *x; \quad // 5$
 $\text{cout} \ll \&x; \quad // \text{address of } x (\text{on } 61 \& 8 \& 10)$
- ↳ $x++; \quad // \text{increment } x \text{ by 4 as } x \text{ stores address of } \text{int} (4 \text{ bytes})$
 $\text{cout} \ll x; \quad // (\text{on } 61 \& 235) \quad // \text{a address increased by 4}$
- ↳ $\text{cout} \ll \text{sizeof}(x); \quad // 8 \quad // \text{size of pointer}$

Main Ideas, Questions & Summary: ↳ $\text{sizeof}(*x); \quad // 4$

★ Bad practice { Segmentation fault } ⇒

↳ `int *ptr;`

`cout << *ptr; // garbage value`

↳ `int *ptr = 0;`

`cout << *ptr; // address not found`

★ Copy Pointer ⇒

↳ `int a = 5;`

`int *ptr1 = &a;`

`int *ptr2 = ptr1; // copy pointer`

`cout << *ptr1 << *ptr2; // 5 5`

★ int array pointer ⇒

↳ `int arr[] = { 52, 145, 78, 136, 23 };`

`int *ptr = arr; // for array pointer don't use (&)`

`int *x = &arr[2]; // in this use (&)`

`cout << ptr; // (0x61f230)`

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

- ↳ cout << "All are same base address";
 cout << arr; // (0x61f230)
 cout << &arr; // (0x61f230)
 cout << arr[0] // (0x61f230)

- ↳ cout << arr+2; // increase address by 2, not value.
 (0x61f238)

- ↳ cout << arr[0]; // 52
 cout << *arr; // 52
 cout << 0[arr]; // 52

- ↳ cout << arr[1]; // 145 [converted into *(arr+1)]
 cout << *(arr+1); // 145
 cout << 1[arr]; // 145

Main Ideas, Questions & Summary:

* char array pointer =>

- ↳ For char array pointer, cout behaviour is different.
- ↳

```
char ch[10] = "MayurSoni";  
char * c = ch;  
  
cout << c; // Instead of ch address it will print whole string  
// MayurSoni
```
- ↳

```
cout << ch; // MayurSoni  
cout << &ch; // 0x618e36  
cout << ch[0]; // M
```
- ↳

```
cout << &c; // 0x618e28  
cout << *c; // M [converted into *(c+0)]  
cout << c; // MayurSoni
```
- ↳

```
cout << c+5; // Soni (start from 5 to end)
```
- ↳

```
char chch = 'R';  
char * cptr = &chch;  
cout << cptr; // print some garbage value along with 'R' as we don't  
// have '10'.  
// (R abc)
```
- ↳

```
char * temp = "Badbeer"; // Bad practice  
cout << temp; // Badbeer
```

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Pointers with Functions \Rightarrow

- ↳ Size in $\text{f}(n)$ is always 8 because all is passed in a pointer & pointer is of 8 size always as it stores address.
- ↳ when we pass array in $\text{f}(n)$, only the address of arr is passed not whole arr is passed.

```

void size [ int arr[] ){
    cout << sizeof(arr); // 8
}

int main(){
    int arr [] = { 4, 28, 50, 1, 100 };
    cout << sizeof(arr); // 20
    size(arr); // 8
}

```

Main Ideas, Questions & Summary: _____

★ Pointers in function ⇒

↳ void change (int *p) {
 cout << p; // (0x618e44)
 // p = p + 10; (Not give desire O/P)
 *p = *p + 10;
}

```
int main() {  
    int a = 5;  
    int *ptr = &a;  
    cout << a; // 5  
    cout << &a; // 0x618e44  
    change (ptr);  
    cout << a; // 15  
}
```

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Double pointer =>

↳ `int a = 5;`
`int *x = &a;`
`int **y = &x;`

`cout << *x; // 5`
`cout << &a; // (memory)`
`cout << *y; // (memory)`
`cout << **y; // 5`

* Reference Variable =>

- ↗ Cannot create null reference variable
 ↗ Pointers are hard to understand, so use this.

↳ `int a = 5;`
`int &b = a; // same memory location but diff name.`
`cout << a << b; // 5 5`
`a++, b++;`
`cout << a << b; // 7 7`

Main Ideas, Questions & Summary:

* Reference variable function ⇒

↳ void solve (int &value)
value += 10;

↳ int main () {
int a = 5;
cout << a; // 5
solve(a);
cout << a; // 15
}

* Program ⇒

↳ int *ptr = 0;
int a = 10;
 $*\text{ptr} = \text{a};$ // give error as ptr is pointing to NULL i.e. 00000000
address which does not exist.

cout << *ptr;

* Program ⇒

↳ char ch = 'a';
char *ptr = &ch;
ch++; // ch will be increment by '1' i.e. to
cout << *ptr; // So

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Program =>

↳ int a = 7;

int * c = &a;

c = c + 1; // address which is stored in c will be incremented by 4 as int.

cout << a << *c; // 7 6422088 (Random)

★ Program =>

↳ int a[5];

int * c;

cout << size(a) << sizeof(*c); // 20 8

★ Program =>

↳ int a[] = {1, 2, 3, 4};

cout << *(a) << *(a+1); // 1 2

Main Ideas, Questions & Summary:

★ Program ⇒

↳ `int a[] = {1, 2, 3, 4};`

`int *p = a++;` // constant pointer so error

// `int *p = (a+1);` we can do this instead

`Cout << *p;`

★ Program ⇒

↳ `int arr[] = {4, 5, 6, 7};`

`int *p = (arr+1);` // point to 5

`Cout << *arr + 9;` // (4+9) 13

★ Program ⇒

↳ `char a[] = "xyz";`

`char *c = &a[0];`

`Cout << c;` // xyz

★ Program ⇒

↳ `char s[] = "hello";`

`char *p = s;`

`Cout << s[0] << p[0];` // h h

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Program =>

```

↳ char arr[20];
int i;
for (i=0; i<10; i++)
    *(arr+i) = 65 + i;
*(arr+i) = '\0';
cout << arr; // ABCDEFGHIJ

```

* Program =>

```

↳ char * ptr;
char str[] = "abcdefghijklmnopqrstuvwxyz";
ptr = str;
ptr = str+5;
cout << ptr; // fgh

```

Main Ideas, Questions & Summary:

Library / Website Ref.: -

★ Program ⇒

↳ int numbers [5];

int *p;

p = numbers;

*p = 10; // num[0]

p = &numbers[2];

*p = 20; // num[2]

p--;

*p = 30; // num[1]

p = numbers + 3;

*p = 40; // num[3]

p = numbers;

*(p+4) = 50; // num[40]

cout << numbers[0...4]; // 10 30 20 40 50

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Program =>

```
↳ char st[] = "ABCD";
for (i=0; st[i] != '\0'; i++) {
    cout << st[i] << *(st)+i << *(i+st) << endl;
}
```

```
|| A 65 A A
B 66 B B
C 67 C C
D 68 D D
```

★ Program =>

```
↳ float arr[5] = { 12.5, 10.0, 13.5, 90.5, 0.5 };
float *ptr1 = &arr[0];
float *ptr2 = ptr + 3;

cout << *ptr2; // 90.5
cout << ptr2 - ptr1; // 3
```

Main Ideas, Questions & Summary:

* Program \Rightarrow

\hookrightarrow void fun (int a[])
cout \ll a[0]; // 2

\hookrightarrow int main () {
int a[] = {1, 2, 3, 4, 5};
fun(a+1);
cout \ll a[0]; // 1
}

* Program \Rightarrow

\hookrightarrow void square (int *p) {

int a = 10;
p = &a;
*p = (*p) * (*p);
}

\hookrightarrow int main () {

int a = 999;
square(&a);
cout \ll a; // 999

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Program \Rightarrow

$\hookrightarrow a = 10;$

$*p = \&a;$

$**q = \&p;$

$b = 20;$

$*q = \&b;$

$(*p)++;$

$cout << a << b; \quad // 10 \quad 21$

* Program \Rightarrow

$\hookrightarrow int f(x, *py, **ppz) \{$

int y, z;

$**ppz += 1;$

$z = **ppz;$

$*py += 2;$

$y = *py;$

$x += 3;$

return x+y+z;

}

$\hookrightarrow int main() \{$

int x, *y, **a;

$c = 4;$

$b = \&c;$

$a = \&y;$

$cout << f(c, b, a); \quad // 19$

}

Main Ideas, Questions & Summary:

Program \Rightarrow

\hookrightarrow int ***r, **q, *p, i=8;

$p = \&i;$

$(*p)++;$

$q = \&p;$

$(**q)++;$

$r = \&q;$

cout << *p << **q << ***r; // 10 10 10

Program \Rightarrow

\hookrightarrow void increment (int **p)

$(*p)++;$

\hookrightarrow int main () {

 int num = 10;

 int *ptr = #

 increment (*ptr);

 cout << num; //

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Count Primes $\Rightarrow \# LC = 204$ (Sieve of Eratosthenes)

↳ Optimized approach

↳ Check theory on gitHub

↳ int CountPrimes (int n) {

 if (n==0) return 0;

 vector<bool> primeArray (n, true);

 int count=0;

 for (i=2; i < n; i++) { // for (i=2; i< sqrt(n); i++)

 if (primeArray[i]) {

 count++;

 int table = 2 * i; // table = i * i

 while (table < n) {

 primeArray[table] = false;

 table += i;

}

}

 return count;

★ Segmented Sieve also

Main Ideas, Questions & Summary:

* HCF/GCD \Rightarrow

↳ Highest common factor / Greatest common factor

↳ $\text{HCF}(a, b) \Rightarrow a > b, \text{HCF}(a-b, b)$
 $\Rightarrow a < b, \text{HCF}(b-a, a)$

↳ int $\text{HCF}(a, b)$ {

 if ($a == 0$) return b ;

 if ($b == 0$) return a ;

 while ($a > 0 \&& b > 0$) {

 if ($a > b$)

$a = a - b$;

 else

$b = b - a$;

}

 return $a == 0 ? b : a$;

}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* LCM \Rightarrow

↳ Lowest common factor

*
$$\text{LCM} = (a * b) / \text{HCF}(a, b)$$

↳ int LCM(a, b){

 return (a * b) / HCF(a, b);

}

* Fast Exponential \Rightarrow

↳ If even $\Rightarrow a^b = (a^{b/2})^2$

↳ If odd $\Rightarrow a^b = (a^{b/2})^2 \cdot a$

↳ int fast (num, pow){

 int ans = 1;

 while (pow > 0){

 if (pow % 1)

 ans = ans * num;

 num = num * num;

 pow = pow / 2;

}

 return ans;

Main Ideas, Questions & Summary:

★ Factorial RE =>

↳ int factorial (n) {
 if (n == 1) return 1;
 return n * factorial (n-1);
}

★ Reverse Count RE =>

↳ void reverse (n) {
 if (n == 0) return;
 cout << n;
 reverse (n-1);
}

★ Ascending Count =>

↳ void asc (n) {
 if (n == 0) return;
 asc (n-1);
 cout << n;
}

★ Fibonacci RE =>

↳ int fibo (n) {
 if (n == 1) return 0;
 if (n == 2) return 1;
 return fibo (n-1) + fibo (n-2);
}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Print Array RE \Rightarrow

↳ Method 1 \Rightarrow

```
↳ void print( arr[], size, index ) {
    if (index >= size) return;
    cout << arr[index];
    print( arr, size, index + 1 );
}
```

↳ Method 2 \Rightarrow

```
↳ void print( arr[], size ) {
    if (size == 0) return;
    cout << arr[0];
    print( arr + 1, size - 1 );
}
```

Main Ideas, Questions & Summary:

* Max in array RE \Rightarrow

↳ void max (arr[], size, index, int & maxi) {
 if (index >= size)
 return;
 if (arr [index] > maxi)
 maxi = arr [index];
 arrayMax (arr, size, index + 1, maxi);
}

* Print digit RE \Rightarrow

↳ void print (n) {
 if (n == 0) return;
 print (n / 10);
 cout << n % 10;
}

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Sorted check RE =>

↳ `bool(vec, int &size, index){`

`if (index == size - 1) return true;`

`if (vec[index] > vec[index + 1]) return false;`

`return Sortcheck(vec, size, index + 1);`

}

★ Binary Search RE =>

↳ `int Search(vec, start, end, target){`

`if (start > end)`

`return -1;`

`int mid = start + (end - start) / 2;`

`if (vec[mid] == target)`

`return mid;`

`if (vec[mid] > target)`

`return Search(vec, start, mid - 1, target);`

`else`

`return Search(vec, mid + 1, end, target);`

}

Main Ideas, Questions & Summary:

★ String check RE =>

↳ Method 1 =>

```
↳ bool string find (name, size , find , index){  
    if (index >= size)  
        return false;  
    if (name [index] == find)  
        return true;  
    return stringfind (name, size, find, index+1);  
}
```

↳ Method 2 =>

```
↳ int string find (name , size , find , index){  
    if (index >= size)  
        return -1;  
    if (name [index] == find)  
        return index;  
    return stringfind (name, size, find, index+1);  
}
```