**GOLDMAN SACHS**

*Day 1*

1. **Given an array of strings, return all groups of strings that are anagrams.**

vector<vector<string> > Anagrams(vector<string>& string\_list) {

//code here

if(string\_list.size()==1 && string\_list[0]=="")

return {{string\_list[0]}};

unordered\_map<string, vector<string>> pp;

for(int i=0;i<string\_list.size();i++){

string tp=string\_list[i];

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

pp[tp].push\_back(string\_list[i]);

}

vector<vector<string>> res;

for(auto it=pp.begin();it!=pp.end();it++){

res.push\_back((\*it).second);

}

return res;

}

1. **O**[**verlapping rectangles**](https://practice.geeksforgeeks.org/problems/overlapping-rectangles1924/1/)

int doOverlap(int L1[], int R1[], int L2[], int R2[]) {

// code here

if(L1[0]>R2[0] || L2[0]>R1[0])

return 0;

if(L1[1]<R2[1] || L2[1]<R1[1])

return 0;

return 1;

}

1. [**Count the subarrays having product less than ‘k**](https://practice.geeksforgeeks.org/problems/count-the-subarrays-having-product-less-than-k1708/1/)**’**

int countSubArrayProductLessThanK(const vector<int>& a, int n, long long k) {

long long int res=0,j=0,i=0,pro=1;

while(i<n){

pro\*=a[i];

while(pro>=k)

pro/=a[j++];

res+=(i-j+1);

i++;

}

return res;

}

*Day 2*

1. [**Given a string, Your task is to complete the function encode that returns the run-length encoded string for the given string.**](https://practice.geeksforgeeks.org/problems/run-length-encoding/1/)

string encode(string src)

{

//Your code here

int i;

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

int c=1;

while(i<src.size()-1 && src[i]==src[i+1]){

c++; i++;

}

cout<<src[i]<<c;

}

}

1. [**Program to find Nth Ugly Number.**](https://practice.geeksforgeeks.org/problems/ugly-numbers2254/1/)

ull getNthUglyNo(int n) {

// code here

ull arr[n];

ull ia=0,ib=0,ic=0, m2=2,m3=3,m5=5,next=1;

arr[0]=1;

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

next=min(min(m2,m3),m5);

arr[i]=next;

if(next==m2){

ia+=1;

m2=arr[ia]\*2;

}

if(next==m3){

ib+=1;

m3=arr[ib]\*3;

}

if(next==m5){

ic+=1;

m5=arr[ic]\*5;

}

}

return next;

}

1. [**Given two strings str1 and str2. We say that str2 divides str1 if it's possible**](https://leetcode.com/problems/greatest-common-divisor-of-strings/) **t**[**o concatenate multiple str2 to get str1. For example, ab divides abab.**](https://leetcode.com/problems/greatest-common-divisor-of-strings/) **I**[**f str2 does not divide str1, return -1. Otherwise, return the smallest string**](https://leetcode.com/problems/greatest-common-divisor-of-strings/)[**str3 such that str3 divides both str1 and str2.**](https://leetcode.com/problems/greatest-common-divisor-of-strings/)

string gcdOfStrings(string str1, string str2) {

if(str1+str2==str2+str1)

return str1.substr(0,\_\_gcd(str1.size(),str2.size()));

else

return "";

}

*Day 3*

1. **F**[**ind the kid which gets tha damaged toy**](https://www.geeksforgeeks.org/distributing-m-items-circle-size-n-starting-k-th-position/) **/ Find the position of M-th item**

int findPosition(int N , int M , int K) {

// code here

if(N==1)

return 1;

int res=M%N+K-1;

if(res==N)

return res;

else

return res%N;

}

1. [**Total Decoding Messages**](https://practice.geeksforgeeks.org/problems/total-decoding-messages1235/1/)

int CountWays(string s){

// Code here

int mod = 1e9 + 7;

int n=s.length();

if(n<1 || s[0]=='0')

return 0;

if(n==1)

return 1;

int c1=1,c2=1;

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

int d=s[i]-'0';

int dd=(s[i-1]-'0')\*10+d;

int c=0;

if(d>0) c+=c2%mod;

if(dd>=10 && dd<=26) c+=c1%mod;

c1=c2;

c2=c;

}

return c2%mod;

}

1. [**Given a pattern containing only I's and D's. I for increasing and D for decreasing.Devise an algorithm to print the minimum number following**](https://practice.geeksforgeeks.org/problems/number-following-a-pattern3126/1)[**that pattern.**](https://practice.geeksforgeeks.org/problems/number-following-a-pattern3126/1)

string printMinNumberForPattern(string S){

// code here

string res;

stack<int> s;

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

s.push(i+1);

if(S.length()==i || S[i]=='I')

while(!s.empty()){

res+=to\_string(s.top());

s.pop();

}

}

return res;

}

*Day 4*

1. **Find max 10 numbers in a list having 10M entries.**

LINK NOT FOUND, but I guess it can be solved by using priority queue in order to reach the solution.

1. [**Given an unsorted array Arr of size N of positive integers. One number**](https://practice.geeksforgeeks.org/problems/find-missing-and-repeating2512/1/)[**'A' from set {1, 2, …N} is missing and one number 'B'**](https://practice.geeksforgeeks.org/problems/find-missing-and-repeating2512/1/)[**occurs twice in array. Find these two numbers.**](https://practice.geeksforgeeks.org/problems/find-missing-and-repeating2512/1/)

int \*findTwoElement(int \*arr, int n) {

// code here

int \*res=new int[2];

sort(arr,arr+n);

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

if(arr[i]==arr[i+1])

res[0]=arr[i];

if(binary\_search(arr,arr+n,i+1))

continue;

else

res[1]=i+1;

}

return res;

}

1. **Find total number of Squares in a N\*N chessboard**

long long int countSquares(int n)

{

return ((n\*(n+1))\*(2\*n+1))/6;

}

1. [**Decode the string**](https://practice.geeksforgeeks.org/problems/decode-the-string2444/1)

string decodedString(string s){

// code here

int n=s.size();

stack <char> st;

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

if(s[i]!=']')

st.push(s[i]);

else {

string res;

while(!st.empty() && st.top()!='['){

res=st.top()+res;

st.pop();

}

st.pop();

string k;

while(!st.empty() && isdigit(st.top())){

k=st.top()+k;

st.pop();

}

int n=stoi(k);

string rep;

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

rep+=res;

for(char c:rep)

st.push(c);

}

}

string final;

while(!st.empty()){

final=st.top()+final;

st.pop();

}

return final;

}

1. [**Minimum Size Subarray Sum**](https://leetcode.com/problems/minimum-size-subarray-sum/)

int minSubArrayLen(int s, vector<int>& nums)

{

int n=nums.size(),res=INT\_MAX,st=0,sum=0;

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

sum+=nums[i];

while(sum>=s){

res=min(res,i+1-st);

sum-=nums[st++];

}

}

return res!=INT\_MAX?res:0;

}

1. [**Array Pair Sum Divisibility Problem**](https://practice.geeksforgeeks.org/problems/array-pair-sum-divisibility-problem3257/1)

bool canPair(vector<int> nums, int k) {

// Code here.

map <int, int> pp;

int r=0,fr=0;

for(auto x : nums)

pp[x%k]++;

if(pp[0] %2 != 0)

return false;

for(auto x : pp)

{ r = x.first;

fr = x.second;

if(r!= 0 && pp[k-r] != fr)

return false;

}

return true;

}