Name-Priyanshu Singh

Albanero Day-2

Question 11:

Q11. Find and Replace Pattern

You have a list of words and a pattern, and you want to know which words in words matches the pattern.

A word matches the pattern if there exists a permutation of letters p so that after replacing every letter x in the pattern with p(x), we get the desired word.

(Recall that a permutation of letters is a bijection from letters to letters: every letter maps to another letter, and no two letters map to the same letter.)

Return a list of the words in words that match the given pattern.

You may return the answer in any order.

Example 1:

Input: words = ["abc","deq","mee","aqq","dkd","ccc"], pattern = "abb"

Output: ["mee","aqq"]

Explanation: "mee" matches the pattern because there is a permutation {a -> m, b -> e, ...}.

"ccc" does not match the pattern because {a -> c, b -> c, ...} is not a permutation,

since a and b map to the same letter.

Note:

1 <= words.length <= 50

1 <= pattern.length = words[i].length <= 20

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin>>n;

string a[n];

cin>>ws;

int j;

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

{

cin>>ws;

cin>>a[j];

}

int k,l,j1;

cin>>ws;

string pattern;

cin>>pattern;

cout<<"Input array : ";

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

cout<<a[i]<<" ";

cout<<endl;

cout<<"Pattern : ";

cout<<pattern<<endl;

vector<long long int>c;int u=1,w;

for(w=0;w<pattern.length();w++)

c.push\_back(-1);

for(j=0;j<pattern.length();j++)

{

for(k=j;k<pattern.length();k++)

{

if(pattern[j]==pattern[k] && c[k]==-1)

c[k]=u;

}

u++;

}

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

{

vector<long long int>b;u=0;

for(w=0;w<a[j].length();w++)

b.push\_back(-1);

for(k=0;k<a[j].length();k++)

{

u++;

for(l=k;l<a[j].length();l++)

{

if(a[j][k]==a[j][l] && b[l]==-1)

{

b[l]=u;

}

}

}

if(b.size()==c.size())

{

for(j1=0;j1<b.size();j1++)

{

if(b[j1]==c[j1])

;

else

break;

}

if(j1==b.size())

cout<<a[j]<<" ";

}

}

}

Question 12:

Q12. Increasing Decreasing String

Given a string s. You should re-order the string using the following algorithm:

Pick the smallest character from s and append it to the result.

Pick the smallest character from s which is greater than the last appended character to the result and append it.

Repeat step 2 until you cannot pick more characters.

Pick the largest character from s and append it to the result.

Pick the largest character from s which is smaller than the last appended character to the result and append it.

Repeat step 5 until you cannot pick more characters.

Repeat the steps from 1 to 6 until you pick all characters from s.

In each step, If the smallest or the largest character appears more than once you can choose any occurrence and append it to the result.

Return the result string after sorting s with this algorithm.

Example 1:

Input: s = "aaaabbbbcccc"

Output: "abccbaabccba"

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

string a;

cin>>a;

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

char j;int k,p,f=0;

string h=a;

while(f<a.length())

{

for(j='a';j<='z';j++)

{

for(k=0;k<a.length();k++)

{

if(a[k]==j)

{

cout<<j;

a[k]=-1;

f++;

break;

}

}

if(f==a.length())

break;

}

for(p=a.length()-1;p>=0;p--)

h=h+a[p];

if(f==a.length())

break;

}

}

Q13. Count the number of 1s in a given 2D array.

  Examples

    [

      [1, 0],

      [0, 0]

    ] ➞ 1

    [

      [1, 1, 1],

      [0, 0, 1],

      [1, 1, 1]

    ] ➞ 7

    [

      [1, 2, 3],

      [0, 2, 1],

      [5, 7, 33]

    ] ➞ 2

Code:

Question 13:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int m,n;

cin>>m>>n;

int a[m][n];

int i,j,f=0;

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

{

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

{

cin>>a[i][j];

if(a[i][j]==1)

f++;

}

}

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

{

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

{

cout<<a[i][j];

}

cout<<endl;

}

cout<<f<<"\n";

}

Question 14:

Q14. Takes an input array of numbers and return a new array, sorted in ascending order (smallest to biggest).

  If the function's argument is null, an empty array, or undefined; return an empty array.

  Examples

    [1, 2, 10, 50, 5] ➞ [1, 2, 5, 10, 50]

    [80, 29, 4, -95, -24, 85] ➞ [-95, -24, 4, 29, 80, 85]

    [] ➞ []

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin>>n;

int a[n];

int j;

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

{

cin>>a[j];

}

int k;int p=0;

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

{

for(k=j+1;k<n;k++)

{

if(a[j]>a[k])

{

p=a[j];

a[j]=a[k];

a[k]=p;

}

}

}

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

{

cout<<a[j]<<" ";

}

}

Question 15:

Q15. Take an input, array of numbers and return only the even values.

  Examples

    [1, 2, 3, 4, 5, 6, 7, 8] ➞ [2, 4, 6, 8]

    [43, 65, 23, 89, 53, 9, 6] ➞ [6]

    [718, 991, 449, 644, 380, 440] ➞ [718, 644, 380, 440]

  Notes

    Return all even numbers in the order they were given.

    All test cases contain valid numbers ranging from 1 to 3000.

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin>>n;

int a[n];

int j;

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

{

cin>>a[j];

if(a[j]%2==0)

cout<<a[j]<<" ";

}

}

Question 16:

Q16. Take input, an array of positive and negative numbers. Return an array where the first element is the count of positive numbers and the second element is the sum of negative numbers.

  Examples

    [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -11, -12, -13, -14, -15] ➞ [10, -65]

    // There are a total of 10 positive numbers.

    // The sum of all negative numbers equals -65.

    [92, 6, 73, -77, 81, -90, 99, 8, -85, 34] ➞ [7, -252]

    [91, -4, 80, -73, -28] ➞ [2, -105]

    [] ➞ []

  Notes

    If given an empty array, return an empty array: []

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin>>n;

int a[n];

int j;

int h=0,d=0;

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

{

cin>>a[j];

if(a[j]>0)

h++;

if(a[j]<0)

d=d+a[j];

}

cout<<h<<" "<<d;

}

Question 17:

Q17. Write a program that returns the sum of all even elements in a 2D matrix.

  Examples

    [

      [1, 0, 2],

      [5, 5, 7],

      [9, 4, 3]

    ] ➞ 6

    // 2 + 4 = 6

    [

      [1, 1],

      [1, 1]

    ] ➞ 0

    [

      [42, 9],

      [16, 8]

    ] ➞ 66

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin>>n;

int m;

cin>>m;

int a[n][m];

int i,j,f=0;

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

{

for(j=0;j<m;j++)

{

cin>>a[i][j];

if(a[i][j]%2==0)

f=f+a[i][j];

}

}

cout<<f<<endl;

}

Question 18:

Q18. Write a program that finds each factor of a given number n. Your solution should return an array of the number(s) that meet this criteria.

  Examples

    (9) ➞ [1, 3, 9]

    // 9 has three factors 1, 3 and 9

    (12) ➞ [1, 2, 3, 4, 6, 12]

    (20) ➞ [1, 2, 4, 5, 10, 20]

    (0) ➞ []

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin>>n;

int a[n];int j,k=0;

for(j=1;j<=n;j++)

{

if(n%j==0)

{

a[k]=j;k++;

}

}

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

cout<<a[j]<<" ";

}

Question 19:

Q19.  Write a program that takes an array of numbers arr and a number n. Return true if the sum of any two elements is equal to the given number. Otherwise, return false.

  Examples

    ([10, 12, 4, 7, 9, 11], 16) ➞ true

    ([4, 5, 6, 7, 8, 9], 13) ➞ true

    ([0, 98, 76, 23], 174) ➞ true

    ([0, 9, 7, 23, 19, 18, 17, 66], 39) ➞ false

    ([2, 8, 9, 12, 45, 78], 1) ➞ false

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin>>n;

int k;

cin>>k;

int a[n];

int j;

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

{

cin>>a[j];

}

sort(a,a+n);

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

{

int p=k-a[j];

if(binary\_search(a,a+n,p))

{

cout<<true;

break;

}

}

if(j==n)

cout<<false;

}

Question 20:

Q20 Given an array of 10 numbers, return the maximum possible total made by summing just 5 of the 10 numbers.

  Examples

    [1, 1, 0, 1, 3, 10, 10, 10, 10, 1] ➞ 43

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 100] ➞ 100

    [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] ➞ 40

  Notes

   You can select any 5 numbers from the given array to form maximum possible total.

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin>>n;

int k=0;

int a[n];

int j;

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

{

cin>>a[j];

}

sort(a,a+n,greater<int>());

for(j=0;j<5;j++)

k=k+a[j];

cout<<k;

}