/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online C Compiler.

Code, Compile, Run and Debug C program online.

Write your code in this editor and press "Run" button to compile and execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Merge and sort array in c

#include <stdio.h>

void mergearray(int arr1[],int arr2[],int n1,int n2){

int i=0,j=0,k=0,arr3[100],temp;

while(i<n1){

arr3[k++]=arr1[i++];

}

while(j<n2){

arr3[k++]=arr2[j++];

}

for(i=0;i<n1+n2;i++){

for(j=i+1;j<n1+n2;j++){

if(arr3[i]>arr3[j]){

temp=arr3[i];

arr3[i]=arr3[j];

arr3[j]=temp;

}

}

}

for(i=0;i<n1+n2;i++){

printf("%d",arr3[i]);

}

}

int main()

{

int arr1[]={1,3,5,7};

int arr2[]={2,4,6,8};

int n1=sizeof(arr1)/sizeof(arr1[0]);

int n2=sizeof(arr2)/sizeof(arr2[0]);

mergearray(arr1,arr2,n1,n2);

return 0;

}

//find largest element in array

#include<stdio.h>

int main(){

int arr[100],n;

printf("Enetr the size of array");

scanf("%d",&n);

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

scanf("%d",&arr[i]);

int max=arr[0];

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

if(arr[i]>max){

max=arr[i];

}

}

printf("%d",max);

return 0;

}

MERGE SORT ARRAY IN C

//Merge sort array

#include<stdio.h>

int main(){

int arr1[100],arr2[100],temp[100],n1,n2,i,j,k=0;

scanf("%d %d",&n1,&n2);

for(i=0;i<n1;i++)

scanf("%d",&arr1[i]);

for(i=0;i<n2;i++)

scanf("%d",&arr2[i]);

i=0,j=0;

while(i<n1&&j<n2){

if(arr1[i]<arr2[j]){

temp[k++]=arr1[i];

i++;

}else

{

temp[k++]=arr2[j];

j++;

}

}

while(i<n1){

temp[k++]=arr1[i];

i++;

}

while(j<n2){

temp[k++]=arr2[j];

j++;

}

for(int i=0;i<n1+n2;i++){

printf("%d",temp[i]);

}

return 0;

}

Input:5

1,2,3,4,5

Ouput: 1,4,3,2,5

#include<stdio.h>

#include<string.h>

int main()

{

char str[100],ch;

int i,len,s,pos,k;

scanf("%s", &str);

len=strlen(str);

int number=0;

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

{

if((str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z'))

{

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

{

printf("%c",ch);

}

ch=str[i];

number=0;

}

else

{

number=number\*10+(str[i]-'0');

}

}

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

{

printf("%c",ch);

}

printf("\n");

}

Input=a1b2c3

Output=abbccc

#include<stdio.h>

#include<string.h>

int main(){

char a[100],temp;

scanf("%s",&a);

int i,j,n=0;

for(int i=0;i<strlen(a);i++){

if(a[i]>='0'&&a[i]<='9'){

temp=a[i-1];

while(a[i]>='0'&&a[i]<='9'){

n=n\*10+(a[i]-'0');

i++;

}

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

printf("%c",temp);

}

}

n=0;

}

}output=a1b2

Aab

#include<stdio.h>

#include<string.h>

int main(){

int i,j,n,mid;

scanf("%d",&n);

mid=(n/2)+1;

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

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

if(i==j){

if(i<=mid){

printf("%d",j);

}else{

printf("%d",n-i+1);

}

}else if(i+j==n+1){

if(i<mid){

printf("%d",(n-i+1));

}else{

printf("%d",i);

}

}

else{

printf(" ");

}

}

printf("\n");

}

}X pattern print

5

1 5

2 4

3

4 2

5 1

MERGE TWO SORTED ARRAY

#include<stdio.h>

int main(){

int a[100],b[100],c[100],m,n,k=0;

scanf("%d %d",&m,&n);

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

scanf("%d",&a[i]);

}

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

scanf("%d",&b[j]);

}

int i=0,j=0;

while(i<m&&j<n){

if(a[i]<b[j]){

c[k++]=a[i];

i++;

}else {

c[k++]=b[j];

j++;

}

}

while(i<m){

c[k++]=a[i];

i++;

}

while(j<n){

c[k++]=b[j];

j++;

}

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

if(i==0)

printf("%d",c[i]);

else

printf(",%d",c[i]);

}

return 0;

}

Output

/tmp/JF7BEpnNJc.o

5

5

1 2 3 4 5

1 2 3 4 5

1,1,2,2,3,3,4,4,5,5

Character between vowels reverse

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

isVowel(char ch){

ch=tolower(ch);

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

}

int main(){

char a[100];

scanf("%s",&a);

int n=strlen(a);

for(int i=n-1;i>=0;i--){

if(isVowel(a[i])){

printf("%s\n",&a[i]);

a[i]='\0';

}

}

}

OUTPUT:

APPLE

E

APPL

/tmp/afm4YkuOaT.o

String permutation

abcd

abcd abdc acbd acdb adcb adbc bacd badc bcad bcda bdca bdac cbad cbda cabd cadb cdab cdba dbca dbac dcba dcab dacb dabc string permutation

// Online C compiler to run C program online

#include <stdio.h>

void permutation(char \*str,int start,int end){

if(start==end)

printf("%s ",str);

else{

for(int i=start;i<=end;i++){

str[start]=str[start]+str[i]-(str[i]=str[start]);

permutation(str,start+1,end);

str[start]=str[start]+str[i]-(str[i]=str[start]);

}

}

}

int main() {

char a[100];

int len;

scanf("%s",&a);

for(len=0;a[len];len++);

// printf("%d",len);

permutation(a,0,len-1);

return 0;

}

Sort array according to frequency of array

#include<stdio.h>

int main(){

int a[100]={2 ,2 ,3 ,4 ,5, 12, 2, 3, 3, 3, 12};

int fr[100]={1,1,1,1,1,1,1,1,1,1,1};

int res[100];

int result[100],count=0,temp;

for(int i=0;i<11;i++){

if(fr[i]==0)

continue;

for(int j=i+1;j<11;j++){

if(fr[j]==0)

continue;

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

fr[i]++;

fr[j]=0;

}

}

}

for(int i=0;i<11;i++){

if(fr[i]!=0){

for(int j=i+1;j<11;j++){

ifca(fr[i]<fr[j]){

temp=fr[i];

fr[i]=fr[j];

fr[j]=temp;

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

}

for(int i=0;i<11;i++){

while(fr[i]>0){

res[count++]=a[i];

fr[i]--;

}

printf("%d ",res[i]);

}

}

3 3 3 3 2 2 2 12 12 5 4

#include<stdio.h>

int calc(int px,int \*py,int \*\*pz){

int y,z;

\*\*pz+=1;

y=\*\*pz;

\*py+=\*\*pz;

y=\*py;

px+=\*py;

return px+y+z;

}

int main(){

int a=5;

int \*b,\*\*c;

b=&a;

c=&b;

printf("%d",calc(a,b,c));

return 0;

}

#include <stdio.h>

int f(int n,int k){

if(n==0)

return 0;

else if(n%2)

return f(n/2,2\*k)+k;

else

return f(n/2,-2\*k)+k;

}

int main(){

printf("%d",f(50,2));

return 0;

}

// Online C compiler to run C program online

#include <stdio.h>

int main() {

// Write C code here

int a,b,temp;

scanf("%d %d",&a,&b);

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

temp=i\*i;

if(temp>a&&temp<=b){

printf("%d ",temp);

}

}

return 0;

}

Output: /tmp/oKN7r4RfLi.o

-1 100

1 4 9 16 25 36 49 64 81 100

**REVERSE FLOYD PATTERN USING NUMBER C**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online C Compiler.

Code, Compile, Run and Debug C program online.

Write your code in this editor and press "Run" button to compile and execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main()

{

int n,temp;

scanf("%d",&n);

int ch;

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

ch=n-i;

temp=i;

printf("%d ",i);

while(ch>0){

temp=temp+ch+i;

printf("%d ",temp);

ch--;

}

printf("\n");

}

return 0;

}

Output

/tmp/oKN7r4RfLi.o

5

1 6 10 13 15

2 7 11 14

3 8 12

4 9

5

**MATRIX ADDITION**

// Online C compiler to run C program online

#include <stdio.h>

int main() {

int n=2,m=2;// Write C code here

int a[2][2]={{1,2},{3,4}},b[2][2]={1,2,3,4};

int c[n][m];

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

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

c[i][j]=a[i][j]+b[i][j];

}

}

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

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

printf("%d ",c[i][j]);

}

printf("\n");

}

return 0;

}

Output

2 4

6 8

#include<stdio.h>

void main(){

int a[3][3]={2,7,6,9,5,1,4,3,8};

int n=sizeof(a)/sizeof(a[0]);

//printf("%d",n);

int diagonalsum1=0;

int diagonalsum2=0;

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

diagonalsum1=diagonalsum1+a[i][i];

diagonalsum2=diagonalsum2+a[i][n-i-1];

}

if(diagonalsum1!=diagonalsum2){

printf("not magic square");

return;

}

int rowsum;

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

rowsum=0;

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

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

}

if(rowsum!=diagonalsum1){

// printf("%d",rowsum);

printf("Not magic square");

return;

}

}

int colsum;

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

colsum=0;

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

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

}

if(colsum!=diagonalsum2){

printf("Not magic square");

return;

}

}

printf("Magic square");

return 0;

}

/tmp/12OMmFb4D2.o

Magic square

Remove space from string

#include<stdio.h>

int main(){

char a[]="go ku la kr is hn an";

int count=0;

for(int i=0;a[i];i++){

if(a[i]!=' ')

a[count++]=a[i];

}

a[count]='\0';

printf("%s",a);

return 0;

}

NGE

#include<stdio.h>

int main(){

int a[100],n,res;

scanf("%d",&n);

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

scanf("%d",&a[i]);

}

// printf("%d",a[0]);

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

res=-1;

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

if(a[i]<a[j]){

res=a[j];

break;

}

}

printf("%d ",res);

}

return 0;

}

Output

/tmp/12OMmFb4D2.o

4

4 3 2 1

-1 -1 -1 -1

Find leaders to right side

// Online C compiler to run C program online

#include <stdio.h>

int leader(int a[],int len){

int i,j;

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

for(j=i+1;j<len;j++){

if(a[i]<=a[j]){

break;

}

}

if(j==len)

printf("%d ",a[i]);

}

}

int main() {

int a[]={5,4,3,2,1};

int len=\*(&a+1)-a;

//printf("%d",leader(a,len));

leader(a,len);

return 0;

}

**Replace every element with the greatest element on right side**

{16, 17, 4, 3, 5, 2}, then it should be modified to {17, 5, 5, 5, 2, -1}.

#include<stdio.h>

int main(){

int a[]={16,17,4,3,5,2},temp;

int n=\*(&a+1)-a;

int maxright=a[n-1];

a[n-1]=-1;

for(int i=n-2;i>=0;i--){

temp=a[i];

a[i]=maxright;

if(maxright<temp){

maxright=temp;

}

}

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

printf("%d ",a[i]);

}

}

Given a boolean matrix mat[M][N] of size M X N, modify it such that if a matrix cell mat[i][j] is 1 (or true) then make all the cells of ith row and jth column as 1.

**Examples:**

***Input:****{{1, 0},  
           {0, 0}}****Output:****{{1, 1}  
              {1, 0}}****Input:****{{0, 0, 0},  
            {0, 0, 1}}****Output:****{{0, 0, 1},  
               {1, 1, 1}}*

***Input:****{{1, 0, 0, 1},  
           {0, 0, 1, 0},  
          {0, 0, 0, 0}}****Output:****{{1, 1, 1, 1},  
               {1, 1, 1, 1},  
              {1, 0, 1, 1}}*

// Online C compiler to run C program online

#include <stdio.h>

#include<stdbool.h>

int main() {

int R,C;

scanf("%d %d",&R,&C);

bool a[R][C];

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

for(int j=0;j<C;j++){

scanf("%d",&a[i][j]);

}

}

bool row[R];

bool col[C];

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

row[i]=0;

}

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

col[i]=0;

}

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

for(int j=0;j<C;j++){

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

row[i]=1;

col[i]=1;

}

}

}

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

for(int j=0;j<C;j++){

if(row[i]==1||col[j]==1){

a[i][j]=1;

}

}

}

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

for(int j=0;j<C;j++){

printf("%d ",a[i][j]);

}

printf("\n");

}

return 0;

}

17. Input:

arr[] = {1, 2, 3, 4, 5, 6, 7}, d = 2

Output: 3 4 5 6 7 1 2

**ARRAY ROTATION**

Input: arr[] = {3, 4, 5, 6, 7, 1, 2}, d=2

Output: 5 6 7 1 2 3 4

#include<stdio.h>

int main(){

int a[]={1,2,3,4,5},d=1,p=1;

int len=\*(&a+1)-a;

int last;

while(p<=d){

last=a[0];

for(int i=0;i<len-1;i++){

a[i]=a[i+1];

}

a[len-1]=last;

p++;

}

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

printf("%d ",a[i]);

}

**Check if given String is Pangram or not**

// Online C compiler to run C program online

#include <stdio.h>

#include<stdbool.h>

bool checkpangram(char a[]){

bool mark[26];

int len;

for(int i=0;i<26;i++)

mark[i]=false;

for(len=0;a[len];len++);

int index;

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

if(a[i]>='A'&&a[i]<='Z'){

index=a[i]-'A';

}

else if(a[i]>='a'&&a[i]<='z'){

index=a[i]-'a';

}

else

continue;

mark[index]=true;

}

for(int i=0;i<26;i++){

if(mark[i]==false)

return false;

}

return true;

}

int main() {

char a[]="The quick brown fox jumps over the lazy dog";

if(checkpangram(a)==true){

printf("true");

}else{

printf("false");

}

return 0;

}

True

**Missing characters to make a string**

**Pangram**

// Online C compiler to run C program online

#include <stdio.h>

#include<stdbool.h>

#include<string.h>

int main() {

char a[]="The quick brown fox jumps over the dog";

int len=strlen(a);

bool mark[26]={false};

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

if(a[i]>='a'&&a[i]<='z')

mark[a[i]-'a']=true;

if(a[i]>='A'&&a[i]<='Z')

mark[a[i]-'A']=true;

}

for(int i=0;i<26;i++){

if(mark[i]==false){

printf("%c",i+'a');

}

}

return 0;

}

/tmp/9ZhOBveU3l.o

Alyz

**Removing punctuations from a given string**

**Input :** %welcome' to @geeksforgeek<s

**Output :** welcome to geeksforgeeks

#include<stdio.h>

#include<string.h>

int main(){

char a[]="Welcome???@@##$ to#$% Geeks%$^for$%^&Geeks";

int len=strlen(a);

//printf("%d",len);

int k=0;

char res[len];

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

if((a[i]>='a'&&a[i]<='z')||(a[i]>='A'&&a[i]<='Z')||a[i]==' '){

res[k++]=a[i];

}

}

printf("%s",res);

}

/tmp/7G3qaAmbMX.o

Welcome to GeeksforGeeks

**Fibonocci using forloop**

#include<stdio.h>

int main(){

int f1=0,f2=1;

int nextterm=f1+f2;

int n=10;

printf("%d %d ",f1,f2);

for(int i=3;i<=n;i++){

printf("%d ",nextterm);

f1=f2;

f2=nextterm;

nextterm=f1+f2;

}

}

0 1 1 2 3 5 8 13 21 34

**Count frequency of array elements**

// Online C++ compiler to run C++ program online

#include <iostream>

#include<bits/stdc++.h>

using namespace std;

int main() {

int a[]={1,2,1,2,4,3,4,5,4,6};

int len=\*(&a+1)-a;

unordered\_map<int,int>mp;

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

mp[a[i]]++;

}

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

if(mp[a[i]]!=-1){

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

mp[a[i]]=-1;

}

}

return 0;

}

/tmp/heHfWZPSVn.o

1 2

2 2

4 3

3 1

5 1

6 1

**Frequency of char in string in c++ using unsorted map**

// Online C++ compiler to run C++ program online

#include <iostream>

#include<bits/stdc++.h>

using namespace std;

int main() {

unordered\_map<char,int>d;

string a="geeksforgeeks";

for(char i:a){

d[i]++;

}

for(char i:a){

if(d[i]!=0){

cout<<i<<" "<<d[i]<<endl;

d[i]=0;

}

}

return 0;

}

/tmp/RceIQucBLs.o

g 2

e 4

k 2

s 2

f 1

o 1

r 1

**DIGIT WITH OPERATOR AT THE END IN C/C++**

//#include<iostream.h>

#include<bits/stdc++.h>

using namespace std;

int main(){

string a="5730+-\*";

int first\_op=0;

for(int i=0;a[i];i++){

if(!(a[i]>='0'&&a[i]<='9')){

first\_op=i;

break;

}

}

int ans=a[0]-'0';

//cout<<ans;

int op=first\_op;

for(int i=1;i<first\_op;i++){

if(a[op]=='+'){

ans=ans+a[i]-'0';

}else if(a[op]=='-'){

ans-=a[i]-'0';

}else if(a[op]=='\*'){

ans\*=a[i]-'0';

}

op++;

}

cout<<ans;

}

/tmp/RceIQucBLs.o

9

**REMOVE SPACE FROM GIVEN STRING IN C/C++**

// Online C compiler to run C program online

#include <stdio.h>

int main() {

char a[]="ge e k h h h h h h h s";

int i=0,j=0;

while(a[i]){

if(a[i]!=' ')

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

i++;

}

a[j]='\0';

printf("%s",a);

return 0;

}

Geekhhhhhhhs

**ADD SPACE FROM STRING IF STARTING LETTER IS CAPITAL LETTER**

**#include<bits/stdc++.h>**

**#include<iostream>**

**using namespace std;**

**int main(){**

**string a="ThisIsMyString";**

**for(int i=0;i<a.length();i++){**

**if(a[i]>='A'&&a[i]<='Z'){**

**// a[i]=a[i]+32;**

**// cout<<a[i];**

**if(i!=0){**

**cout<<" ";**

**}**

**cout<<a[i];**

**}**

**else{**

**cout<<a[i];**

**}**

**}**

**}**

/tmp/RceIQucBLs.o

This Is My String

**FLOIYD TRIANGLE ZOHO INTERVIEW QUESTIONS**

// Online C compiler to run C program online

#include <stdio.h>

int main() {

char ca='A',sa='a';

int n,flag=0,count=0;

scanf("%d",&n);

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

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

if(flag==0){

printf("%c",ca+count++);

flag=1;

}else if(flag==1){

printf("%c",sa+count++);

flag=0;

}

if(count>=26)

return;

}

printf("\n");

if((i+1)%2==1)

flag=0;

}

return 0;

}

/tmp/VeDCHMoTBk.o

27

A

bC

DeF

gHiJ

KlMnO

pQrStU

VwXyZ

guess the random number in java

// Online Java Compiler

// Use this editor to write, compile and run your Java code online

import java.util.\*;

class HelloWorld {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Random rand=new Random();

int number=rand.nextInt(10),guess,max=3,countguess=0;

boolean outofguess=false;

while(!outofguess){

if(countguess<max){

System.out.println(max-countguess+" guess remaining \n");

System.out.println("Guess :");

guess=sc.nextInt();

if(guess==number){

System.out.println("You won the game");

System.out.println("The number is "+ number);

break;

}

countguess++;

}else{

System.out.println("You lost the actual number is "+ number);

outofguess=true;

}

}

}

}

Output

java -cp /tmp/RZiVTblTDV HelloWorld

3 guess remaining

Guess :5

2 guess remaining

Guess :

7

1 guess remaining

Guess :

0

You won the game

The number is 0

**ARRAYLIST CUSTOM OBJECTS**

// Online Java Compiler

// Use this editor to write, compile and run your Java code online

import java.util.\*;

class Student{

int id;

String name;

int mark;

Student(Integer id,String name,Integer mark){

this.id=id;

this.name=name;

this.mark=mark;

}

}

class HelloWorld {

public static void main(String[] args) {

List<Student> l1=new ArrayList<>();

l1.add(new Student(1,"GOkul",97));

l1.add(new Student(2,"RAHUL",100));

for(int i=0;i<l1.size();i++){

System.out.println(l1.get(i).id+" "+l1.get(i).name+" "+l1.get(i).mark);

System.out.println();

}

}

}

java -cp /tmp/GoS0A22FDZ HelloWorld

1 GOkul 972 RAHUL 100

**SQUARE PATTERN IN C**

// Online C compiler to run C program online

#include <stdio.h>

int main() {

int n,i,j,size;

scanf("%d",&n);

size=2\*n-1;

int a[size][size];

int first=0,last=size-1;

while(n!=0){

for(i=first;i<=last;i++){

for(j=first;j<=last;j++){

if(i==first||i==last||j==first||j==last){

a[i][j]=n;

}

}

}

first++;

last--;

n--;

}

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

for(j=0;j<=size-1;j++){

printf("%d ",a[i][j]);

}

printf("\n");

}

return 0;

}

/tmp/LQP2NVOTf2.o

5

5 5 5 5 5 5 5 5 5

5 4 4 4 4 4 4 4 5

5 4 3 3 3 3 3 4 5

5 4 3 2 2 2 3 4 5

5 4 3 2 1 2 3 4 5

5 4 3 2 2 2 3 4 5

5 4 3 3 3 3 3 4 5

5 4 4 4 4 4 4 4 5

5 5 5 5 5 5 5 5 5

**CHECK THE DATE IS VALID OR NOT**

// Online C compiler to run C program online

#include <stdio.h>

int isleap(int y){

if((y % 4 == 0) && (y % 100 == 0) && (y % 400 == 0))

return 1;

else

return 0;

}

int isvalid(int d,int m,int y){

if(y<1800||y>9999){

return 0;

}

if(m<1||m>12){

return 0;

}

if(d<1||d>31){

return 0;

}

if(m==2){

if(isleap(y)){

if(d<=29){

return 1;

}else

return 0;

}else{

if(d<=28){

return 1;

}else{

return 0;

}

}

}

if(m==4||m==6||m==9||m==11){

if(d<=30)

return 1;

else

return 0;

}

return 1;

}

int main() {

int d,m,y;

printf("Enter the date in dd/mm/yyyy format");

scanf("%d/%d/%d",&d,&m,&y);

if(isvalid(d,m,y)){

printf("valid date");

}else{

printf("invalid date");

}

return 0;

}

/tmp/ppIaSYywoH.o

Enter the date in dd/mm/yyyy format29/02/2003

invalid date

**FIND FIRST AND SECOND LARGEST ELEMT IN ARRAY IN C/C++**

#include<stdio.h>

int main(){

int a[]={10,10,10,11};

int f\_largest=0,s\_largest=-1;

int len=\*(&a+1)-a;

//printf("%d",len);

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

if(a[i]>a[f\_largest]){

f\_largest=i;

}

}

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

if(a[i]!=a[f\_largest]){

if(s\_largest==-1)

s\_largest=i;

else if(a[i]>a[s\_largest])

s\_largest=i;

}

}

printf("first max=%d second largest=%d",a[f\_largest],a[s\_largest]);

}

Output

/tmp/az1xdho8Ej.o

first max=11 second largest=10

**CHECK THE NUMBER IS PERFECT NHUMBER OR NOT**

// Online C compiler to run C program online

#include <stdio.h>

int main() {

int a=28,rem,sum=0,i=1;

while(i<a){

if(a%i==0)

sum=sum+i;

i++;

}

// for(int i=1;i<a;i++){

// rem=a%i;

// //printf("%d",rem);

// if(rem==0)

// sum=sum+i;

// }

// printf("%d",sum);

if(sum==a)

printf("perfect number");

else

printf("not a perfect number");

return 0;

}

Output

/tmp/cklK6DKAHN.o

perfect number

sort the array with points

5 points if a perfect square

4 points if multiple by 4 and 6

3 points if it is even number;

input :[10,36,3,49,12];

ouput:[36,1,49,20,6];

// Online C compiler to run C program online

#include <stdio.h>

int perfectsquare(int n){

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

if((n%i==0)&&(n/i==i)){

return 1;

}

}

return 0;

}

int main() {

int a[]={8,25,2,81,32};

int len=\*(&a+1)-a;

int point[len];

//printf("%d",len);

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

int count=0;

if(perfectsquare(a[i])){

count+=5;

}

if(a[i]%4==0&&a[i]%6==0)

count+=4;

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

count+=3;

point[i]=count;

}

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

for(int j=0;j<len;j++){

if(point[j]<point[j+1]){

int temp=point[j];

point[j]=point[j+1];

point[j+1]=temp;

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}else if(point[j]==point[j+1]){

if(a[j]<a[j+1]){

int temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

}

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

printf("%d ",a[i]);

}

// printf("%d",count);

// if(perfectsquare(num)){

// printf("%d is perfeect square",num);

// }

return 0;

}

Output

/tmp/cklK6DKAHN.o

81 25 32 8 2

**DIAMOND PATTERN WITH NUMBER IN C**

#include<stdio.h>

int main(){

int n,space,i,j,c,k=1,temp;

scanf("%d",&n);

space=n-1;

int res=0;

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

for(c=1;c<=space;c++){

printf(" ");

}

space--;

// k++;

//k=0,temp=0

temp=i-1;

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

res=k+++temp---j;

printf("%d ",res);

// k++;

// temp--;

}

//k++;

printf("\n");

}

space=1;

//printf("%d",k);

k--;

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

for(c=1;c<space;c++){

printf(" ");

}

space++;

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

printf("%d ",k);

k-=1;

}

printf("\n");

}

}

Output

n=5;

1

3 2

6 5 4

6 5 4

3 2

1

**N QUEEN PROBLEM**

**// Online C compiler to run C program online**

**#include <stdio.h>**

**#include<stdlib.h>**

**int count=0;**

**int safe(char \*\*mat,int n,int row,int col){**

**int i,j;**

**for( i=row;i>=0;i--){**

**if(mat[i][col]=='Q')**

**return 0;**

**}**

**for( i=row,j=col;i>=0&&j>=0;i--,j--){**

**if(mat[i][j]=='Q')**

**return 0;**

**}**

**for( i=row,j=col;i>=0&&j>=0;i--,j++){**

**if(mat[i][j]=='Q')**

**return 0;**

**}**

**return 1;**

**}**

**void queen(char\*\* mat,int n,int row){**

**if(row==n){**

**printf("%d\n",++count);**

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

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

**printf("%c ",mat[i][j]);**

**}**

**printf("\n");**

**}**

**printf("\n\n\n");**

**}**

**else{**

**for(int col=0;col<n;col++){**

**if(safe(mat,n,row,col)==1){**

**mat[row][col]='Q';**

**queen(mat,n,row+1);**

**mat[row][col]='-';**

**}**

**}**

**}**

**}**

**int main() {**

**char \*\*mat;**

**int n,row,col;**

**scanf("%d",&n);**

**mat=(char\*\*)malloc(n\*sizeof(char\*));**

**for(row=0;row<n;row++){**

**mat[row]=(char\*)malloc(n\*sizeof(char));**

**}**

**for(row=0;row<n;row++){**

**for(col=0;col<n;col++){**

**mat[row][col]='-';**

**}**

**}**

**queen(mat,n,0);**

**return 0;**

**}**

/tmp/M391mOPhaz.o

5

1

Q - - - -

- - Q - -

- - - - Q

- Q - - -

- - - Q -

2

Q - - - -

- - - Q -

- Q - - -

- - - - Q

- - Q - -

3

- Q - - -

- - - Q -

Q - - - -

- - Q - -

- - - - Q

4

- Q - - -

- - - - Q

- - Q - -

Q - - - -

- - - Q -

5

- - Q - -

Q - - - -

- - - Q -

- Q - - -

- - - - Q

6

- - Q - -

- - - - Q

- Q - - -

- - - Q -

Q - - - -

7

- - - Q -

Q - - - -

- - Q - -

- - - - Q

- Q - - -

8

- - - Q -

- Q - - -

- - - - Q

- - Q - -

Q - - - -

9

- - - - Q

- Q - - -

- - - Q -

Q - - - -

- - Q - -

10

- - - - Q

- - Q - -

Q - - - -

- - - Q -

- Q - - -

**EQULIBRIUM INDEX OF THE ARRAY**

#include<stdio.h>

int equilibrium(int a[],int n){

int lsum, rsum;

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

lsum=0;

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

lsum+=a[j];

rsum=0;

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

rsum+=a[j];

if(lsum==rsum){

return i;

}

}

return -1;

}

int main(){

int a[]={-7,1,5,2,-4,3,0};

int len=\*(&a+1)-a;

printf("%d",equilibrium(a,len));

}

/tmp/M391mOPhaz.o

3

**SUM AND DIFFERENCE OF TWO NUMBERS USING C POINTER**

#include <stdio.h>

#include<stdlib.h>

void update(int \*a,int \*b) {

     int c=\*a+\*b;

    int d=abs(\*a-\*b);

    \*a=c;

    \*b=d;

    // Complete this function

}

int main() {

    int a, b;

    int \*pa = &a, \*pb = &b;

    scanf("%d %d", &a, &b);

    update(pa, pb);

    printf("%d\n%d", a, b);

    return 0;

}

/tmp/xxQ18ewxaF.o

4 5

9

1

**REVERSE THE ARRAY USING Malloc dma**

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int num, \*arr, i;

    scanf("%d", &num);

    arr = (int\*) malloc(num \* sizeof(int));

    for(i = 0; i < num; i++) {

        scanf("%d", arr + i);

    }

    /\* Write the logic to reverse the array. \*/

    for(i = num-1; i>=0; i--)

        printf("%d ", \*(arr + i));

    return 0;

}

Input (stdin)

Download

* **6**
* **16 13 7 2 1 12**

Your Output (stdout)

* **12 1 2 7 13 16**

.Given an array of integers, compute the maximum value for each integer in the index, by either summing all the digits or multiplying all the digits. (Choose which

#include<stdio.h>

int main()

{

int a[]={120 ,24 ,71 ,10 ,59};

int num, rem, prod = 1,sum,temp;

// printf("Enter a number: ");

// scanf("%d", &num);

for(int i=0;i<5;i++){

num=a[i],sum=0,prod=1;

while(num>0){

temp=num%10;

sum=sum+temp;

num=num/10;

}

//printf("%d",sum);

num=a[i];

while(num != 0)

{

rem = num % 10; // get the last-digit

prod \*= rem; // calculate product of digits

num /= 10; // remove the last digit

}

if(sum>prod){

printf("%d ",sum);

}

else{

printf("%d ",prod);

}

// printf("%d ",prod);

}

// printf("%d", prod);

return 0;

}

/tmp/4BLmoGnyk6.o

3 8 8 1 45

5. Given a matrix of 2D array of n rows and m coloumns. Print this matrix in ZIG-ZAG fashion as shown in figure.

Example:

Input:

1 2 3

4 5 6

7 8 9

Output:

1 2 4 7 5 3 6 8 9

#include<stdio.h>

int main()

{

int matrix[3][3] = {1,2,2

,3,4,5

,6,7,8};

int M = 3, N = 3;

int result[M\*N];

result[0] = matrix[0][0]; //Initialization start position

int i=0, j=0, k=1;

while(k < N\*M)

{

//move up-right first

while(i>=1&&j<N-1){

i--;

j++;

printf("%d",i);

result[k++] = matrix[i][j];

}

//when we can't move up-right ,then move right one step

if(j<N-1){

j++;

result[k++] = matrix[i][j];

}

//if we can't move right,just move down one step

else if(i<M-1) {

i++;

result[k++] = matrix[i][j];

}

//After that,we will move down-left until it can't move

while(i<M-1&&j>=1) {

i++;

j--;

result[k++] = matrix[i][j];

}

//if we can't move down-left,then move down

if(i<M-1){

i++;

result[k++] = matrix[i][j];

}

//if we can't move down,just move right

else if(j<N-1){

j++;

result[k++] = matrix[i][j];

}

}

for (int i=0; i<M\*N; i++)

// printf("%d ",result[i]) ;

return 0;

}

operation gives the maximum value)

|  |
| --- |
| 21.Given N. print the following snake pattern (say N = 4). condition: must not use arrays ( 1D array or 2D array like Matrix ). |
|  | 1 2 3 4 |
|  | 8 7 6 5 |
|  | 9 10 11 12 |
|  | 16 15 14 13 |

#include<stdio.h>

int main()

{

int n,k=0;

scanf("%d",&n);

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

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

if(i%2==1){

k++;

printf("%d ",k);

}else{

printf("%d ",k);

k--;

}

}k+=n;

printf("\n");

}

}

/tmp/MGnL9LpqlC.o

4

1 2 3 4

8 7 6 5

9 10 11 12

16 15 14 13

|  |
| --- |
| 22.Given N. print the Latin Matrix (say N = 3). condition: must not use strings(aka character literals), arrays (both 1D and 2D), inbuilt functions(like rotate). |
|  |  |
|  |  |
|  |  |
|  | A B C |
|  | B C A |
|  | C A B |

#include <stdio.h>

#include <math.h>

int main()

{

int k=1,n=3;

// int i,j,k=1,n;

// int range[n];

// printf("Enter the number for the Latin Square: ");

// scanf("%d",&n);

// printf("Enter the numbers from 1-n: \n");

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

// printf("Enter: ");

// scanf("%d", &range[n]);

// }

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

printf("\n");

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

{

printf("%c ",k+'A'-1);

if (k==n)

k=1;

else

k++;

}

k++;

}

return(0);

}

/tmp/zRMp00ZM0F.o

A B C

B C A

C A B

List custom object and class in java

// Online Java Compiler

// Use this editor to write, compile and run your Java code online

import java.util.\*;

class Student{

int id;

String name;

Student(int id,String name){

this.id=id;

this.name=name;

}

public void printdetails(){

System.out.println("id:"+this.id);

System.out.println("Name:"+this.name);

}

// public String toString(){

// return "Name:"+this.name+"id:"+this.id;

// }

}

class HelloWorld {

public static void main(String[] args) {

List <Student> l1=new ArrayList<>();

Scanner sc=new Scanner(System.in);

System.out.println("Enter number of details");

int n=sc.nextInt();

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

String name=sc.next();

l1.add(new Student(i,name));

}

for(Student s:l1){

s.printdetails();

}

// System.out.println("Hello, World!");

}

}

23. mobile keypad numbr format in c

input :2

output : a,b,c

// Online C compiler to run C program online

#include <stdio.h>

#include<string.h>

void lettercombination(int n2,int n1){

char\* b[]={"0","1","abc","def","ghi","jkl","mno","pqrs","tuv","wxyz"};

char\* s1[100];

char\* s2[100];

s1[0]=b[n2];

s2[0]=b[n1];

char res[100];

int len1=strlen(s1[0]);

int len2=strlen(s2[0]);

int k=0;

for(int i=0;i<len1;i++){

for(int j=0;j<len2;j++){

printf("%c %c, ",s1[0][i],s2[0][j]);

}

}

for(int i=0;i<strlen(res);i++){

printf("%c",res[i]);

}

}

int main() {

char a[][10]={"0","1","a,b,c","d,e,f","g,h,i","j,k,l","m,n,o","p,q,r,s","t,u,v","w,x,y,z"};

int n;

printf("Enter the number");

scanf("%d",&n);

if(n>1&&n<9){

switch(n){

case 2:printf("%s",a[2]);break;

case 3:printf("%s",a[3]);break;

case 4:printf("%s",a[4]);break;

case 5:printf("%s",a[5]);break;

case 6:printf("%s",a[6]);break;

case 7:printf("%s",a[7]);break;

case 8:printf("%s",a[8]);break;

case 9:printf("%s",a[9]);break;

}

}else if(n>20&&n<=99){

int rem=n%10,que=n/10;

if(rem!=0&&rem!=1&&que!=1){

int n1=rem,n2=que;

lettercombination(n2,n1);

}else{

printf("invalid number");

}

}else{

printf("Invalid number");

}

return 0;

}

/tmp/8t7wyaq9Lt.o

Enter the number22

a a, a b, a c, b a, b b, b c, c a, c b, c c

**REMOVE ELEMENTS FROM THE ARRAY**

/tmp/0MrdiWCsM7.o

Enter the value to remove:2

1 3 4 5

// Online C compiler to run C program online

#include <stdio.h>

int main() {

int a[]={1,2,3,4,5};

int len=\*(&a+1)-a,val,index=0;

//printf("%d",len);

printf("Enter the value to remove:");

scanf("%d",&val);

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

if(a[i]!=val){

a[index++]=a[i];

}

}

len--;

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

printf("%d ",a[i]);

return 0;

}

,

**REMOVE DUPLICATE ELEMENT FROM SORTED ARRAY**

// Online C compiler to run C program online

#include <stdio.h>

int main() {

int a[]={1, 2, 2, 3, 4, 4, 4, 5, 5,6,6};

int len=\*(&a+1)-a;

int j=0;

for(int i=0;i<len-1;i++){

if(a[i]!=a[i+1]){

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

}

}

a[j++]=a[len-1];

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

printf("%d ",a[i]);

}

return 0;

}

/tmp/gbR2SUi8VM.o

1 2 3 4 5 6

**REMOVE DUPLICATE ELEMENT FROM ARRAY**

**// Online C compiler to run C program online**

**#include <stdio.h>**

**int main() {**

**int a[]={1,2,1,3,4,2,5,3};**

**int len=\*(&a+1)-a,k=0;**

**for(int i=0;i<len;i++){**

**for(int j=i+1;j<len;j++){**

**if(a[i]==a[j]){**

**for(k=j;k<len;k++){**

**a[k]=a[k+1];**

**}**

**len--;**

**j--;**

**}**

**}**

**}**

**for(int i=0;i<len;i++)**

**printf("%d ",a[i]);**

**return 0;**

**}**

/tmp/vBJ0t6OUzT.o

1 2 3 4 5

**MOVE ALL ZEROS AT END OF THE ARRAY**

// Online C compiler to run C program online

#include <stdio.h>

int main() {

int a[]={1,2,0,3,4,0,0,5,1,0,0};

int len=\*(&a+1)-a;

int j=0;

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

if(a[i]!=0){

int temp=a[i];

a[i]=a[j];

a[j]=temp;

j++;

}

}

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

printf("%d ",a[i]);

}

return 0;

}

/tmp/vBJ0t6OUzT.o

1 2 3 4 5 1 0 0 0 0 0

**COONCATETATE ARRAY USING MALLOC**

/\*\*

 \* Note: The returned array must be malloced, assume caller calls free().

 \*/

int\* getConcatenation(int\* nums, int numsSize, int\* returnSize){

\*returnSize=numsSize\*2;

int \* ans=(int\*)malloc(sizeof(int)\*(numsSize\*2));

int i=0;

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

    ans[i]=nums[i];

    ans[i+numsSize]=nums[i];

}

return ans;

}

Input : 1,2,3

Output: 1,2,3,1,2,3

 Running Sum of 1d Array

class Solution {

    public int[] runningSum(int[] nums) {

        int res=0,k=0;

        int n=nums.length;

        int[] ans=new int[n];

        int i=1;

        while(i<=n){

            res=0;

            for(int j=0;j<i;j++){

                res=res+nums[j];

            }

            i++;

            ans[k++]=res;

        }

        return ans;

    }

}

**INPUT:1,2,3,4**

**OUTPUT: 1,3,6,10**

**METHOD 2:**

class Solution {

public int[] runningSum(int[] nums) {

int[] ans=new int[nums.length];

ans[0]=nums[0];

for(int i=1;i<nums.length;i++)

ans[i]=ans[i-1]+nums[i];

return ans;

}

}

C PROGRAMMING LANGUAGE

/\*\*

\* Note: The returned array must be malloced, assume caller calls free().

\*/

int\* runningSum(int\* nums, int numsSize, int\* returnSize){

\*returnSize=numsSize;

int\* sum=(int\*)malloc(sizeof(int)\*numsSize);

int i=1,res=0,k=0;

while(i<=numsSize){

res=0;

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

res+=nums[j];

}

sum[k++]=res;

i++;

}

return sum;

}

/\*\*

 \* Note: The returned array must be malloced, assume caller calls free().

 \*/

int\* runningSum(int\* nums, int numsSize, int\* returnSize){

    \*returnSize=numsSize;

    int\* sum=(int\*)malloc(sizeof(int)\*numsSize);

    for(int i=0;i<numsSize;i++){

        int res=0;

        for(int j=0;j<=i;j++){

            res+=nums[j];

        }

        sum[i]=res;

    }

    return sum;

}

1920. Build Array from Permutation

class Solution {

    public int[] buildArray(int[] nums) {

        int n=nums.length;

        // int k=0;

        // int[] ans=new int[nums.length];

        for(int i=0;i<n;i++){

            nums[i]=n\*(nums[nums[i]]%n)+nums[i];

        }

        for(int i=0;i<n;i++){

            nums[i]=nums[i]/n;

        }

        return nums;

    //    return ans;

    }

}

**Output: 0,1,2,4,5,3**

**Explaination:** [**https://medium.com/theleanprogrammer/build-array-from-permutation-907ff395eb40**](https://medium.com/theleanprogrammer/build-array-from-permutation-907ff395eb40)

TWO SUM IN C

/\*\*

\* Note: The returned array must be malloced, assume caller calls free().

\*/

int\* twoSum(int\* nums, int numsSize, int target, int\* returnSize){

\*returnSize=2;

int temp;

int\* ans=(int\*)malloc(sizeof(int)\*2);

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

temp=target-nums[i];

for(int j=i+1;j<numsSize;j++){

if(temp==nums[j]){

ans[0]=i;

ans[1]=j;

}

}

}

return ans;

}

TWO SUM IN C++

class Solution {

public:

vector<int> twoSum(vector<int>& nums, int target) {

vector<int>ans;

int temp;

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

temp=target-nums[i];

for(int j=i+1;j<nums.size();j++){

if(nums[j]==temp){

ans.push\_back(i);

ans.push\_back(j);

}

}

}

return ans;

}

};

TWO SUM IN JAVA

class Solution {

public int[] twoSum(int[] nums, int target) {

int[] res=new int[2];

Map<Integer,Integer> mp=new HashMap<>();

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

if(mp.containsKey(target-nums[i])){

res[1]=i;

res[0]=mp.get(target-nums[i]);

return res;

}

else mp.put(nums[i],i);

}

return res;

}

}

Final Value of Variable After Performing Operations

int finalValueAfterOperations(char \*\* operations, int operationsSize){

    int x=0;

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

        int pinc=strcmp(operations[i],"X++");

        int poinc=strcmp(operations[i],"++X");

        int pdec=strcmp(operations[i],"X--");

        int podec=strcmp(operations[i],"--X");

        if(pinc==0||poinc==0){

            x=x+1;

        }else if(pdec==0||podec==0){

            x=x-1;

        }

    }

    return x;

}

**Input:** operations = ["--X","X++","X++"]

**Output:** 1

int finalValueAfterOperations(char \*\* operations, int operationsSize){

    int x=0;

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

        x+=(operations[i][1]=='+')?1:-1;

    }

    return x;

}

1672. Richest Customer Wealth

class Solution {

    public int maximumWealth(int[][] accounts) {

        int m=accounts.length;

        int n=accounts[0].length;

        int k=0;

        int sum=0;

        int max=0;

        for(int i=0;i<m;i++){

            sum=0;

            for(int j=0;j<n;j++){

                sum=sum+accounts[i][j];

            }

            if(max<=sum){

                max=sum;

            }

        }

        return max;

    }

}

**Input:** accounts = [[1,2,3],[3,2,1]]

**Output:** 6

1672. Richest Customer Wealth in c

int maximumWealth(int\*\* accounts, int accountsSize, int\* accountsColSize){

int max=0;

int sum=0;

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

sum=0;

for(int j=0;j<\*accountsColSize;j++){

sum+=accounts[i][j];

}

if(max<=sum){

max=sum;

}

}

return max;

}

2114. Maximum Number of Words Found in Sentences

class Solution {

    public int mostWordsFound(String[] sentences) {

        int max=0;

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

            String[] newString=sentences[i].split("\s+");

            max=Math.max(newString.length,max);

        }

        return max;

    }

}

**Input:** sentences = ["alice and bob love leetcode", "i think so too", "this is great thanks very much"]

**Output:** 6

class Solution {

    public int mostWordsFound(String[] sentences) {

        int max=0;

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

                String sentence=sentences[i];

                String[] newString=sentence.split(" ");

                int count=newString.length;

                if(count>=max){

                    max=count;

                }

            }

            return max;

    }

}

2114. Maximum Number of Words Found in Sentences in c

int countsentence(char\* str){

int len=strlen(str);

if(len==0)

return 0;

int count=0;

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

if(str[i]==' '){

count++;

}

}

return count+1;

}

int mostWordsFound(char \*\* sentences, int sentencesSize){

int max=0,sum=0;

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

sum=0;

sum=countsentence(sentences[i]);

if(sum>=max){

max=sum;

}

}

return max;

}

7. Queries  
R A B C  
1 56 67 89  
2 89 54 90  
3 78 91 83  
4 69 72 95  
R – Rollno, A, B, C – Marks in three subjects  
Given the above matrix, print the result of the queries given the following syntax.  
The first input string has a single character denoting the field to be printed.  
\* – All fields of the selected rows  
A – Print only field A  
B – Print only field B  
C – Print only field C

The second string contains the condition – <field><relational\_operator><value>.  
> – Greater than  
< – Less than  
= – Equal to

I/P: \*  
A>70

O/P:

2 89 54 90  
3 78 91 83  
4 69 72 95

I/P: A  
C<90

O/P:

56  
78

// Online C compiler to run C program online

#include <stdio.h>

#include<stdbool.h>

int main() {

int a[4][4]={{1,56, 67, 89},

{2,89 ,54, 90},

{3,78 ,91, 83},

{4,69 ,72 ,95}

};

printf("\n");

char ch;

printf("\nEnter your choice:");

scanf("%c",&ch);

bool a1=true;

// switch(ch){

// case '\*':

// printf("R A B C ");

// printf("\n");

// for(int i=0;i<4;i++){

// for(int j=0;j<4;j++){

// printf("%d ",a[i][j]);

// }

// printf("\n");

// }

// break;

// case 'A':

// printf("A\n");

// for(int i=0;i<4;i++){

// printf("%d \n",a[i][1]);

// }

// break;

// case 'B':

// printf("B\n");

// for(int i=0;i<4;i++){

// printf("%d \n",a[i][2]);

// }

// break;

// case 'C':

// printf("C\n");

// for(int i=0;i<4;i++){

// printf("%d \n",a[i][3]);

// }

// break;

// }

char s1,s2[100];

printf("Enter two string:");

scanf("%c %s",&s1,s2);

int ans=((s2[2]-'0')\*10+(s2[3]-'0'));

// printf("%d",ans);

switch(s1){

case '\*':

printf("R A B C \n");

if(s2[1]=='>'){

if(s2[0]=='A'){

for(int i=0;i<4;i++){

for(int j=0;j<4;j++){

if(a[i][1]>ans)

printf("%d ",a[i][j]);

}

printf("\n");

}

}else if(s2[0]=='B'){

for(int i=0;i<4;i++){

for(int j=0;j<4;j++){

if(a[i][2]>ans)

printf("%d ",a[i][j]);

}

printf("\n");

}

}else if(s2[0]=='C'){

for(int i=0;i<4;i++){

for(int j=0;j<4;j++){

if(a[i][3]>ans)

printf("%d ",a[i][j]);

}

printf("\n");

}

}

}else if(s2[1]=='<'){

if(s2[0]=='A'){

for(int i=0;i<4;i++){

for(int j=0;j<4;j++){

if(a[i][1]<ans)

printf("%d ",a[i][j]);

}

printf("\n");

}

}else if(s2[0]=='B'){

for(int i=0;i<4;i++){

for(int j=0;j<4;j++){

if(a[i][2]<ans)

printf("%d ",a[i][j]);

}

printf("\n");

}

}else if(s2[0]=='C'){

for(int i=0;i<4;i++){

for(int j=0;j<4;j++){

if(a[i][3]<ans)

printf("%d ",a[i][j]);

}

printf("\n");

}

}

}else if(s2[1]=='='){

}

}

// while(a1){

// switch(ch){

// case '\*':

// printf("R A B C ");

// printf("\n");

// for(int i=0;i<4;i++){

// for(int j=0;j<4;j++){

// printf("%d ",a[i][j]);

// }

// printf("\n");

// }

// break;

// case 'A':

// printf("A");

// for(int i=0;i<4;i++){

// printf("%d \n",a[i][1]);

// }

// break;

// }

// }

// printf("A\n");

// for(int i=0;i<4;i++){

// printf("%d \n",a[i][1]);

// }

// break;

return 0;

}

Difference Between Element Sum and Digit Sum of an Array

int differenceOfSum(int\* nums, int numsSize){

    int sum=0,totaldigit=0;

    for(int i=0;i<numsSize;i++){

        sum+=nums[i];

        int n=nums[i],digitsum=0;

        while(n){

            digitsum+=n%10;

            n/=10;

        }

    totaldigit+=digitsum;

    }

    int diff=abs(sum-totaldigit);

    return diff;

}

Output

[1,15,6,3]

Output

9

class Solution {

public int differenceOfSum(int[] nums) {

int sum=0,digitsum=0;

for(int n:nums){

sum+=n;

String numstring=Integer.toString(n);

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

digitsum+=numstring.charAt(i)-'0';

}

}

return Math.abs(sum-digitsum);

}

}

1313. Decompress Run-Length Encoded List

Input: nums = [1,2,3,4]

Output: [2,4,4,4]

Explanation: The first pair [1,2] means we have freq = 1 and val = 2 so we generate the array [2].

The second pair [3,4] means we have freq = 3 and val = 4 so we generate [4,4,4].

At the end the concatenation [2] + [4,4,4] is [2,4,4,4].

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online C Compiler.

Code, Compile, Run and Debug C program online.

Write your code in this editor and press "Run" button to compile and execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include<stdlib.h>

int main()

{

int a[]={1,1,2,3};

int len=\*(&a+1)-a;

int res[len];

int temp;

int k=0;

for(int i=0;i<len;i+=2){

temp=a[i];

while(temp--){

res[k++]=a[i+1];

}

}

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

printf("%d ",res[i]);

}

return 0;

}

**LOOK AND SAY IN C**

// Online C compiler to run C program online

#include <stdio.h>

int main() {

int temp,count,newnum,k;

long int num=1;

for(int i=0;i<10;i++){

printf("%d \n",num);

newnum=0;

k=1;

while(num!=0){

temp=num%10;

num=num/10;

count=1;

while(temp==num%10){

count++;

num=num/10;

}

newnum=(((count\*10)+temp)\*k)+newnum;

k=k\*100;

}

num=newnum;

}

return 0;

}

Ouput

1

11

21

1211

111221

312211

13112221

1113213211

**ROMAN TO INTEGER IN C**

// Online C compiler to run C program online

#include <stdio.h>

int roman\_to\_integer(char s){

switch(s){

case 'I':

return 1;

case 'V':

return 5;

case 'X':

return 10;

case 'L':

return 50;

case 'D':

return 100;

case 'C':

return 500;

case 'M':

return 1000;

default:

return 0;

}

}

int roman\_to\_int(char \*str){

int int\_num=roman\_to\_integer(str[0]);

for(int i=1;str[i]!='\0';i++){

int prev\_num=roman\_to\_integer(str[i-1]);

int curr\_num=roman\_to\_integer(str[i]);

if(prev\_num<curr\_num){

int\_num=int\_num-prev\_num+(curr\_num-prev\_num);

}else{

int\_num+=curr\_num;

}

}

return int\_num;

}

int main() {

char \*a="DCXXI";

printf("%d",roman\_to\_int(a));

return 0;

}

Input

XX

Output

20

class Solution {

public int romanToInt(String s) {

int n=s.length();

Map<Character,Integer> roman=new HashMap<>();

roman.put('I',1);

roman.put('V',5);

roman.put('X',10);

roman.put('L',50);

roman.put('C',100);

roman.put('D',500);

roman.put('M',1000);

int num=roman.get(s.charAt(n-1));

for(int i=n-2;i>=0;i--){

if(roman.get(s.charAt(i))>=roman.get(s.charAt(i+1))){

num+=roman.get(s.charAt(i));

}else{

num-=roman.get(s.charAt(i));

}

}

return num;

}

}

Roman to integer in c

int romanToInt(char \* s){

int i=0, number=0;

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

char current\_char = s[i];

switch(current\_char){

case 'I':

if(s[i+1]=='V'){

number += 4;

i++;

} else if (s[i+1] == 'X'){

number += 9;

i++;

} else {

number += 1;

}

break;

case 'V':

number += 5;

break;

case 'X':

if(s[i+1]=='L'){

number += 40;

i++;

} else if (s[i+1] == 'C'){

number += 90;

i++;

} else {

number += 10;

}

break;

case 'L':

number += 50;

break;

case 'C':

if(s[i+1]=='D'){

number += 400;

i++;

} else if (s[i+1] == 'M'){

number += 900;

i++;

} else {

number += 100;

}

break;

case 'D':

number += 500;

break;

case 'M':

number += 1000;

break;

}

i++;

}

return number;

}

1678. Goal Parser Interpretation

// Online C compiler to run C program online

#include <stdio.h>

#include<string.h>

int main() {

char command[]="G()(al)";

int len=strlen(command);

char ans[len];

int k=0;

for(int i=0;command[i]!='\0';i++){

if(command[i]=='G'){

ans[k++]='G';

}else if(command[i]=='('){

if(command[i+1]==')'){

ans[k++]='o';

}else{

ans[k++]='a';

ans[k++]='l';

}

}

}

ans[k]='\0';

printf("%s",ans);

return 0;

}

class Solution {

    public String interpret(String command) {

        String ans="";

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

            if(command.charAt(i)=='G'){

                ans+='G';

            }else if(command.charAt(i)=='('){

                if(command.charAt(i+1)==')'){

                    ans+='o';

                }else{

                    ans+="al";

                }

            }

        }

        return ans;

    }

}

Input: G()()al

Output: GOOAL

char \* interpret(char \* command){

int len=strlen(command);

char \*ans=malloc(sizeof(int)\*len+1);

int k=0;

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

// char c=command[i];

if(command[i]=='G'){

ans[k++]=command[i];

}else if(command[i]=='('){

if(command[i+1]==')'){

ans[k++]='o';

}else{

ans[k++]='a';

ans[k++]='l';

}

}

}

ans[k]='\0';

return ans;

}

1528. Shuffle String

char \* restoreString(char \* s, int\* indices, int indicesSize){

    char \*ans=(char\*)malloc(sizeof(char)\*indicesSize+1);

    for(int i=0;i<indicesSize;i++){

        ans[indices[i]]=s[i];

    }

    ans[indicesSize]='\0';

    return ans;

}

class Solution {

public String restoreString(String s, int[] indices) {

char[] ans=new char[indices.length];

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

ans[indices[i]]=s.charAt(i);

}

return new String(ans);

}

}

Input:

**Input:** s = "codeleet", indices = [4,5,6,7,0,2,1,3]

**Output:** "leetcode"

**Explanation:** As shown, "codeleet" becomes "leetcode" after shuffling.

2194. Cells in a Range on an Excel Sheet

Input: s = "K1:L2"

Output: ["K1","K2","L1","L2"]

Explanation:

The above diagram shows the cells which should be present in the list.

The red arrows denote the order in which the cells should be presented.

class Solution {

public List<String> cellsInRange(String s) {

List<String> ans=new ArrayList<>();

char cstart=s.charAt(0);

char cend=s.charAt(3);

char nstart=s.charAt(1);

char nend=s.charAt(4);

for(char i=cstart;i<=cend;i++)

for(char j=nstart;j<=nend;j++){

ans.add(""+i+j);

}

return ans;

}

}

class Solution {

public List<String> cellsInRange(String s) {

List<String> ans=new ArrayList<>();

char cstart=s.charAt(0);

char cend=s.charAt(3);

char nstart=s.charAt(1);

char nend=s.charAt(4);

// char[] str=s.tocharArray();

for(char i=cstart;i<=cend;i++)

for(char j=nstart;j<=nend;j++){

ans.add(String.format("%c%c",i,j));

// ans.add(new String(new char[]{i,j}));

}

return ans;

}

}

Integer to roman

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online C Compiler.

Code, Compile, Run and Debug C program online.

Write your code in this editor and press "Run" button to compile and execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include<string.h>

int main()

{

char I[][10]={"","I","II","III","IV","V","VI","VII","VIII","IX"};

char X[][10]={"","X","XX","XXX","XL","L","LX","LXX","LXXX","XC"};

char L[][10]={"","C","CC","CCC","XC","D","DX","DXX","DXXX","CM"};

char M[][10]={"","M","MM","MMM"};

char roman[20];

int n;

scanf("%d",&n);

strcpy(roman,M[n/1000]);

strcat(roman,L[(n%1000)/100]);

strcat(roman,X[(n%100)/10]);

strcat(roman,I[n%10]);

printf("%s",roman);

// printf("%s",I[n%10]);

return 0;

}

1221. Split a String in Balanced Strings

**Input:** s = "RLRRLLRLRL"

**Output:** 4

**Explanation:** s can be split into "RL", "RRLL", "RL", "RL", each substring contains same number of 'L' and 'R'.

int balancedStringSplit(char \* s){

    int ans=0,l=0;

    for(int i=0;s[i]!='\0';i++){

        if(s[i]=='R'){

            ++l;

        }else{

            --l;

        }

        if(l==0)

        ans++;

    }

    return ans;

}

class Solution {

public int balancedStringSplit(String s) {

int l=0,ans=0;

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

if(s.charAt(i)=='L'){

++l;

}else{

--l;

}

if(l==0){

ans++;

}

}

return ans;

}

}

Spiral matrix in c

1 2 3 4

12 13 14 5

11 16 15 6

10 9 8 7

output: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

// Online C compiler to run C program online

#include <stdio.h>

int main() {

int a[4][4]={{1,2,3,4},

{12,13,14,5},

{11,16,15,6},

{10,9,8,7}

};

int rowstart=0,rowend=sizeof(a)/sizeof(a[0]);

int colstart=0,colend=sizeof(a[0])/rowend;

// printf("%d %d",rowend,colend);

while(rowstart<rowend&&colstart<colend){

for(int i=colstart;i<colend;i++){

printf("%d ",a[rowstart][i]);

}

rowstart+=1;

for(int i=rowstart;i<rowend;i++){

printf("%d ",a[i][colend-1]);

}

colend-=1;

for(int i=colend-1;i>=colstart;i--){

printf("%d ",a[rowend-1][i]);

}

rowend-=1;

for(int i=rowend-1;i>=rowstart;i--){

printf("%d ",a[i][colstart]);

}

colstart+=1;

}

return 0;

}

Chemical bomb pattern in c

/tmp/yOvOyMsAXE.o

I I

CF FC

A I I A

T D D T

I I I I

O C C O

N A N

O C C O

I I I I

T D D T

A I I A

CF FC

I I

// Online C compiler to run C program online

#include <stdio.h>

#include<string.h>

int main() {

// char a[2][100]={"Hydration","acidification"};

char a[]="Acidification";

// int count;

// scanf("%d",&count);

// char a[count][100];

// for(int i=0;i<count;i++)

// scanf("%s",a[i])

int len=strlen(a),mid=(len/2),mid1=mid+1,mid2=mid-1;

// int len2=strlen(a[1])

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

if(a[i]>96){

a[i]=a[i]-32;

}else{

a[i]=a[i];

}

}

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

if(i<=mid){

mid1--,mid2++;

for(int j=0;j<len;j++){

if(j==0||j==len-1){

printf("%c",a[mid2]);

}else if(i==j||i+j==len-1){

printf("%c",a[mid1]);

}else{

printf(" ");

}

// mid1++,mid2--;

}

printf("\n");

}else{

mid1++,mid2--;

for(int j=0;j<len;j++){

if(j==0||j==len-1){

printf("%c",a[mid2]);

}else if(i==j||i+j==len-1){

printf("%c",a[mid1]);

}else{

printf(" ");

}

// mid1++,mid2--;

}

printf("\n");

}

}

/\*

int len;

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

len=strlen(a[0]);

}

printf("%d",len);

\*/ return 0;

}

Inner loop c

// Online C compiler to run C program online

#include <stdio.h>

int main() {

int n;

scanf("%d",&n);

int len=(2\*n)-1;

int a[len][len];

int first=0,last=len-1;

while(n!=0){

for(int i=first;i<=last;i++){

for(int j=first;j<=last;j++){

a[i][j]=n;

}

}

first++;

last--;

n--;

}

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

for(int j=0;j<len;j++){

printf("%d",a[i][j]);

}

printf("\n");

}

return 0;

}

Outpuu

/tmp/6jkMs5OxzM.o

9

99999999999999999

98888888888888889

98777777777777789

98766666666666789

98765555555556789

98765444444456789

98765433333456789

98765432223456789

98765432123456789

98765432223456789

98765433333456789

98765444444456789

98765555555556789

98766666666666789

98777777777777789

98888888888888889

99999999999999999

// Online C compiler to run C program online

#include <stdio.h>

#include<string.h>

#include<stdbool.h>

int main() {

char a[]=" -1234";

int ans=0,i=0;

bool sign=true;

while(a[i]==' '){

i++;

}

if(a[i]=='-'){

sign=false;

i++;

}

for(;a[i]!='\0';++i){

ans=ans\*10+a[i]-'0';

}

if(!sign){

ans=-ans;

}

printf("%d",ans);

return 0;

}

/tmp/jQvZsdJJig.o

-1234

70. Climbing Stairs

Input: n = 2

Output: 2

Explanation: There are two ways to climb to the top.

int climbStairs(int n){

    if(n<=2)

    return n;

    int x=1,y=2;

    for(int i=3;i<n;i++){

        int temp=y;

        y+=x;

        x=temp;

    }

    return x+y;

}

14. Longest Common Prefix

Example 1:

Input: strs = ["flower","flow","flight"]

Output: "fl"

Example 2:

Input: strs = ["dog","racecar","car"]

Output: ""

Explanation: There is no common prefix among the input strings.

char \* longestCommonPrefix(char \*\* strs, int strsSize){

char a;

int i,n=0;

while(a=strs[0][n]){

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

if(strs[i][n]!=a){

strs[0][n]=0;

return strs[0];

}

}

n++;

}

strs[0][n]=0;

return strs[0];

}

class Solution {

public String longestCommonPrefix(String[] strs) {

String prefix=strs[0];

for(int i=1;i<strs.length;i++){

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

if(j==strs[i].length()||prefix.charAt(j)!=strs[i].charAt(j)){

prefix=prefix.substring(0,j);

break;

}

}

}

return prefix;

}

}

1720. Decode XORed Array

**Example 1:**

**Input:** encoded = [1,2,3], first = 1

**Output:** [1,0,2,1]

**Explanation:** If arr = [1,0,2,1], then first = 1 and encoded = [1 XOR 0, 0 XOR 2, 2 XOR 1] = [1,2,3]

/\*\*

 \* Note: The returned array must be malloced, assume caller calls free().

 \*/

int\* decode(int\* encoded, int encodedSize, int first, int\* returnSize){

\*returnSize=encodedSize+1;

int\* arr=(int\*)malloc(sizeof(int)\*(encodedSize+1));

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

arr[0]=first;

arr[i+1]=arr[i]^encoded[i];

}

return arr;

}

class Solution {

    public int[] decode(int[] encoded, int first) {

       int temp=first,k=1;

       int[] ans=new int[encoded.length+1];

       ans[0]=first;

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

           temp=temp^encoded[i];

           ans[k++]=temp;

       }

       return ans;

    }

}

2433. Find The Original Array of Prefix Xor

**Example 1:**

**Input:** pref = [5,2,0,3,1]

**Output:** [5,7,2,3,2]

**Explanation:** From the array [5,7,2,3,2] we have the following:

- pref[0] = 5.

- pref[1] = 5 ^ 7 = 2.

- pref[2] = 5 ^ 7 ^ 2 = 0.

- pref[3] = 5 ^ 7 ^ 2 ^ 3 = 3.

- pref[4] = 5 ^ 7 ^ 2 ^ 3 ^ 2 = 1.

**Example 2:**

**Input:** pref = [13]

**Output:** [13]

**Explanation:** We have pref[0] = arr[0] = 13.

/\*\*

 \* Note: The returned array must be malloced, assume caller calls free().

 \*/

int\* findArray(int\* pref, int prefSize, int\* returnSize){

    \*returnSize=prefSize;

    int \*ans=(int\*)malloc(sizeof(int)\*prefSize);

    ans[0]=pref[0];

    for(int i=1;i<prefSize;i++){

        ans[i]=pref[i]^pref[i-1];

    }

    return ans;

}

class Solution {

    public int[] findArray(int[] pref) {

        int len=pref.length;

        int[] ans=new int[len];

        ans[0]=pref[0];

        for(int i=1;i<len;i++){

            ans[i]=pref[i]^pref[i-1];

        }

        return ans;

    }

}

1313. Decompress Run-Length Encoded List

**Example 1:**

**Input:** nums = [1,2,3,4]

**Output:** [2,4,4,4]

**Explanation:** The first pair [1,2] means we have freq = 1 and val = 2 so we generate the array [2].

The second pair [3,4] means we have freq = 3 and val = 4 so we generate [4,4,4].

At the end the concatenation [2] + [4,4,4] is [2,4,4,4].

**Example 2:**

**Input:** nums = [1,1,2,3]

**Output:** [1,3,3]

/\*\*

\* Note: The returned array must be malloced, assume caller calls free().

\*/

int\* decompressRLElist(int\* nums, int numsSize, int\* returnSize){

int c=0;

for(int i=0;i<numsSize;i+=2)

c+=nums[i];

int \*ans=(int\*)malloc(sizeof(int)\*c);

\*returnSize=c;

for(int i=0,k=0;i<numsSize;i+=2){

int freq=nums[i];

int temp=nums[i+1];

while(freq>0){

ans[k++]=temp;

freq--;

}

}

return ans;

}

class Solution {

public int[] decompressRLElist(int[] nums) {

int freq,temp=0,k=0,c=0;

for(int i=0;i<nums.length;i+=2){

c=c+nums[i];

}

int[] ans=new int[c];

for(int i=0;i<nums.length;i+=2){

freq=nums[i];

temp=nums[i+1];

while(freq>0){

ans[k++]=temp;

freq--;

}

}

return ans;

}

}

**COunting bits in c**

**Input: n = 2**

**Output: [0,1,1]**

**Explanation:**

**0 --> 0**

**1 --> 1**

**2 --> 10**

**Input: n = 5**

**Output: [0,1,1,2,1,2]**

**Explanation:**

**0 --> 0**

**1 --> 1**

**2 --> 10**

**3 --> 11**

**4 --> 100**

**5 --> 101**

class Solution {

    public int[] countBits(int n) {

        int[] ans=new int[n+1];

        ans[0]=0;

        for(int i=1;i<=n;i++)

        ans[i]=ans[i/2]+i%2;

        return ans;

    }

}

/\*\*

\* Note: The returned array must be malloced, assume caller calls free().

\*/

int\* countBits(int n, int\* returnSize){

int\* ans=(int\*)malloc(sizeof(int)\*(n+1));

\*returnSize=n+1;

ans[0]=0;

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

ans[i]=ans[i/2]+i%2;

return ans;

}

Power of 2

bool isPowerOfThree(int n){

    if(n==1)return true;

    if(n<=0) return false;

    while(n>=2){

        if(n%2!=0)return false;

        n=n/2;

    }

    return n==1;

}

**Power of 3**

bool isPowerOfThree(int n){

    if(n==1)return true;

    if(n<=0) return false;

    while(n>=3){

        if(n%3!=0)return false;

        n=n/3;

    }

    return n==1;

}

**Power of 4**

bool isPowerOfThree(int n){

    if(n==1)return true;

    if(n<=0) return false;

    while(n>=4){

        if(n%4!=0)return false;

        n=n/4;

    }

    return n==1;

}

**Balance the paranthesis**

**Input: s = "()"**

**Output: true**

**Input: s = "()[]{}"**

**Output: true**

**Input: s = "(]"**

**Output: false**

class Solution {

public boolean isValid(String s) {

if(s.length()%2!=0)return false;

Stack<Character> st=new Stack();

for(char c:s.toCharArray()){

if(c=='('||c=='{'||c=='['){

st.push(c);

}else if(c==')'&&!st.isEmpty()&&st.peek()=='('){

st.pop();

}else if(c=='}'&&!st.isEmpty()&&st.peek()=='{'){

st.pop();

}else if(c==']'&&!st.isEmpty()&&st.peek()=='['){

st.pop();

}

else {

return false;

}

}

return st.isEmpty();

}

}

Contains duplicate

Input: nums = [1,2,3,1]

Output: true

Input: nums = [1,2,3,4]

Output: false

Input: nums = [1,1,1,3,3,4,3,2,4,2]

Output: true

class Solution {

    public boolean containsDuplicate(int[] nums) {

       Arrays.sort(nums);

       for(int i=0;i<nums.length-1;i++){

           if(nums[i]==nums[i+1])return true;

       }

        return false;

    }

}

**Input: PROGRAM**

**OUTPUT**

**G**

**GR**

**GRA**

**GRAM**

**GRAMP**

**GRAMPR**

**GRAMPRO**

**#include<stdio.h>**

**int main(){**

**char a[]="PROGRAM";**

**int i,j,k,len=0,index;**

**for(i=0;a[i]!='\0';i++){**

**len++;**

**}**

**// printf("%d",len);**

**int mid=len/2,space;**

**for( i=1;i<=len;i++){**

**space=len-i;**

**for(j=1;j<=space;j++){**

**printf(" ");**

**}**

**for(k=0;k<len-space;k++){**

**index=mid+k;**

**if(index>=len)index=index-len;**

**printf("%c",a[index]);**

**}**

**printf("\n");**

**}**

**}**