Write a program to find the median of two arrays. If not sorted, sort the arrays first

int prob\_12\_sort(int \*arr,int n){

int temp;

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

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

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

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;}}

}

}

int prob\_12\_median(int \*arr1,int \*arr2,int n,int m){

prob\_12\_sort(arr1,n);

prob\_12\_sort(arr2,m);

int med1=arr1[n/2];

int med2=arr2[m/2];

int med=(med1+med2)/2;

printf("%d",med);

}

int main(){

int arr1[5]={1,2,3,4,5};

int arr2[6]={6,7,8,9,10,11};

prob\_12\_median(arr1,arr2,5,6);

}

Write a C Program to Convert a Person's Name in Abbreviated Form. If your name is **Ghanendra Pratap Yadav** Then your Abbreviated Form Will be **G. P. Yadav**

void prob\_11(){

    char arr[1000];

    gets(arr);

    int length;

    length=strlen(arr);

    int start=0;

    //printf("a");

    for(int i=0;i<=length;i++){

        if(arr[i]==' '){

            //printf("a");

            printf("%c.",arr[start]);

            start=i+1;

        }

        if(arr[i]=='\0'){

            for(int j=start;j<=(length-1);j++){

                printf("%c",arr[j]);

            }

        }

    }

}

C Program to Split the Array in 3 Parts. Add First and End Elements of Each Sub-array and show the result

int prob\_1(){

    int n;

    scanf("%d",&n);

    int arr[n];

    for(int i=0;i<n;i++){

        scanf("%d",&arr[i]);

    }

    int sub1[3],sub2[3],sub3[n-6];

    for(int i=0;i<3;i++){

        sub1[i]=arr[i];

    }

    for(int i=0;i<3;i++){

        sub2[i]=arr[i+3];

    }

    for(int i=0;i<(n-6);i++){

        sub3[i]=arr[i+6];

    }

    int sum1=sub1[0]+sub1[2];

    int sum2=sub2[0]+sub2[2];

    int sum3=sub3[0]+sub3[2];

    printf("%d %d %d",sum1,sum2,sum3);

}

Write a program to find the power of any number of the array using the function. Take the input of power and number from user.

int prob\_2(int base,int power){

    if(power!=0){

        return(base\*prob\_2(base,power-1));

    }

    else{

        return 1;

    }

}

C Program to Reverse a Sentence Using Recursion

void prob\_6(char \*arr,int start,int end){

    if(start>=end){

        return;

    }

    char temp=arr[start];

    arr[start]=arr[end];

    arr[end]=temp;

    prob\_6(arr,(start+1),(end-1));

}

int main(){

    char chr[1000];

    gets(chr);

    int num=strlen(chr);

    prob\_6(chr,0,(num-1));

    puts(chr);

}

Convert a binary number to decimal and vice-versa using functions. Take input from user, and store the value as array

int prob\_3(int n){

    int rem,arr[100],i=0;

    while(n){

        rem=n%2;

        arr[i]=rem;

        n=n/2;

        i++;

    }

    printf("0");

    for(int j=(i-1);j>=0;j--){

        printf("%d",arr[j]);

    }

}

Convert an octal Number to decimal and vice-versa using functions. Take input from user, and store the value as array

int prob\_4(int n){

      int rem,arr[100];

      int i=0;

    while(n){

        rem=n%8;

        arr[i]=rem;

        n=n/8;

        i++;

    }

    printf("0");

    for(int j=(i-1);j>=0;j--){

        printf("%d",arr[j]);

    }

}

Convert a binary number to octal and vice-versa using functions. Take input from user, and store the value as array.

int prob\_5(int n){

    //binary to decimal

    int m=n;

    int count=0;

    while(n){

        n=n/10;

        count++;

    }

    //printf("%d",count);

    int rem,sum=0,i=0;

    while(m){

        rem=m%10;

        sum+=rem\*prob\_2(2,i);

        m=m/10;

        count--;

        i++;

    }

    printf("decimal number is: %d\n",sum);

    printf("octal number of the number is:");

    prob\_4(sum);

}

Write a C program to find sum of all even or odd numbers in given range using recursion.

int prob\_7\_even(int start,int end){

    if(start>end){

        return 0;

    }

    if(start%2==0){

        return(start+prob\_7\_even(start+2,end));

    }

    else{

        return(prob\_7\_even(start+1,end));

    }

}

int prob\_7\_odd(int start,int end){

    if(start>end){

        return 0;

    }

    if(start%2!=0){

        return(start+prob\_7\_odd(start+2,end));

    }

    else{

        prob\_7\_odd(start+1,end);

    }

}

Write a C program to find sum of alternate elements of array using recursion.

int prob\_8(int \*arr,int n,int i){

if(i>=n){

return 0;

}

if(i%2==0){

return(arr[i]+prob\_8(arr,n,i+2));

}

else{

prob\_8(arr,n,i+1);

}

}

int main(){

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

printf("%d",prob\_8(arr,5,0));

}

Write a program to exchange two numbers of the array using functions. Let the user decide the numbers to be exchanged. Show the result of before and after exchange.

int prob\_9(){

    int n;

    scanf("%d",&n);

    int arr[n];

    for(int i=0;i<n;i++){

        scanf("%d",&arr[i]);

    }

    printf("ORIGINAL ARRAY\n");

    for(int i=0;i<n;i++){

        printf("%d ",arr[i]);

    }

    printf("\n");

    int a,b;

    printf("enter the index that you want to swap\n");

    printf("\n");

    scanf("%d %d",&a,&b);

    int temp;

    temp=arr[a];

    arr[a]=arr[b];

    arr[b]=temp;

    printf("NEW ARRAY\n");

     for(int i=0;i<n;i++){

        printf("%d ",arr[i]);

    }

}

Write a program that extracts and adds the two least significant digits of any number. Define three functions addTwoDigits(), lastDigit() and secondLastDigit(). The main function should call addTwoDigits(). The lastDigit() and secondLastDigit() functions should be called from function addTwoDigits()

int prob\_10\_secondlastDigits(int m){

    return((m/10)%10);

}

int prob\_10\_lastDigit(int m){

    return(m%10);

}

int prob\_10\_add\_twoDigits(int m){

    return(prob\_10\_lastDigit(m)+prob\_10\_secondlastDigits(m));

}

Write a program to find the longest common substring of two strings

int prob\_13(char \*chr1,char \*chr2){

    int a=strlen(chr1);

    int b=strlen(chr2);

    for(int i=0;i<a;i++){

        for(int j=0;j<b;j++){

            if((chr1[i]==chr2[j] && chr1[i+1]==chr2[j+1]) ||(chr1[i]==chr2[j] && chr1[i-1]==chr2[j-1])){

                printf("%c",chr1[i]);

            }

        }

    }

}

int main(){

    char chr1[100], chr2[100];

    gets(chr1);

    gets(chr2);

    prob\_13(chr1,chr2);

}