Functions

LEVEL 1

1. Advika is trying

```
#include <stdio.h>
int NccCells(int x,int y);
int main()
{
    int x,y;
     scanf("%d %d",&x,&y);
    printf("%d",NccCells(x,y));
            return 0;
}
int NccCells(int x,int y){
    return ((x+1)/2)*((y+1)/2);
      }
```

2. Simon

```
#include <stdio.h>
#include<math.h>
int convertBinarytoOctal(long long int );
int main()
{ long long num;
  scanf("%lld",&num);
  convertBinarytoOctal(num);
            return 0;
}
int convertBinarytoOctal(long long binaryNumber)
{ int oct=0,dec=0,i=0;
  while(binaryNumber!=0)
  { dec+=(binaryNumber%10)*(pow(2,i));
    i++;
    binaryNumber/=10;
  }
  i=1;
  while(dec!=0)
  { oct+=(dec%8)*i;
    dec/=8;
    i*=10;
  }
  printf("%d",oct);
  return 0;
}
```

3. Issac is a

```
#include <stdio.h>
int convert(int);
int main()
{ int ndays;
  scanf("%d",&ndays);
  convert(ndays);
          return 0;
}
int convert(int ndays)
{ int y,w;
  y=ndays/365;
  ndays-=y*365;
  w=ndays/7;
  ndays-=w*7;
  printf("%d Years %d Weeks %d Days",y,w,ndays);
  return 0;
}
```

4. Sajid is an 8th

```
#include <stdio.h>
int facto(int n)

{
   if(n==0)
   return 1;
   else
   return(n*facto(n-1));
}
int main()
{
int n;
long fact;
scanf("%d",&n);
fact=facto(n);
printf("%ld",fact);
return 0;
}
```

5. Selvan asks

```
#include<string.h>
#include<stdio.h>
int isISBN(char isbn[])
{
          int n = strlen(isbn);
          if (n != 10)
                    return 0;
          int i,sum = 0;
```

```
        for (i = 0; i < 9; i++)
        {
                int digit = isbn[i] - '0';
                if (0 > digit || 9 < digit)
                        return 0;
                sum += (digit * (10 - i));
        }
        char last = isbn[9];
        if (last != 'X' && (last < '0' || last > '9'))
                return 0;

        sum += ((last == 'X') ? 10 :(last - '0'));

        return (sum % 11 == 0);
}
int main()
{
  int t;
  scanf("%d",&t);
  while(t--)
  {
            char isbn[1000];
     scanf("%s",isbn);
     if (isISBN(isbn))
                printf("Valid\n");
            else
                printf("Invalid\n");
  }
        return 0;
}
```

6. Simon is wasting

```
#include <stdio.h>
float bill(int);
int main()
{ int unit;
  scanf("%d",&unit);
  bill(unit);
        return 0;
}
float bill(int unit)
{ float amt;
  if(unit<=50)
   amt=unit*0.50;
  else if(unit>50&&unit<=150)
   amt=25+((unit-50)*0.75);
  else if(unit>150&&unit<=250)
   amt=100+((unit-150)*1.2);
  else
   amt=220+((unit-250)*1.5);
```

```
  printf("%0.2f",amt);
  return 0;
}
```

7. Simon celebrates

```
#include <stdio.h>
int leap(int);
int main()
{ int y;
  scanf("%d",&y);
  leap(y);
          return 0;
}
int leap(int y)
{ if(y%4!=0)
   printf("Not a Leap Year");
  else
   printf("Leap Year");
  return 0;
}
```

8. Simon wants

```
#include <stdio.h>
#include <math.h>
int isPerfectSquare(long long x)
{

    if(x>=0)
    {
        float sqr=sqrt(x);
        return (sqr*sqr==x);
    }
    else
        return 0;
}
int main()
{
    long long num;
    scanf("%lld",&num);
    if(isPerfectSquare(num))
        printf("NO");
    else
        printf("YES");
            return 0;
}
```

9. Hassan gets a job

```
#include <stdio.h>
```

```
#include <stdlib.h>
int numind(int n)
{
   if(n==100) return 1;
   int rem=n%10;
   n=n/10;
   return rem+n;
}
int main()
{
   int n;
   scanf("%d",&n);
   int arr[n],sum=0,i,j,k;
   for(i=0;i<n;i++)
   {
      scanf("%d",&arr[i]);
      sum+=(i+1)*numind(arr[i]);
   }
   printf("Weight of given input sequence=%d\n",sum);
   int found=0;
   for(i=n-1;i>=0;--i)
      for(j=i-1;j>=0;--j)
         if(arr[i]<arr[j])
         {
            found=1;
            int temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
            for(k=0;k<n;++k)
               printf("%d ",arr[k]);
            printf("\n");
         }
   int sum1=0;
   for(i=0;i<n;++i)
      sum1+=(i+1)*numind(arr[i]);
   if(found==1)
      printf("Maximum sequence weight=%d",sum1);
         return 0;
}
```

10. Tina is a Bachelor

```
#include <stdio.h>
int sum(int arr[],int start,int len);
int main()
{ int arr[20],N,sumofarray,i;
  scanf("%d",&N);
  for( i=0;i<N;i++)
   scanf("%d",&arr[i]);
 sumofarray=sum(arr,0,N);
 printf("%d",sumofarray);
```

```
        return 0;
}
int sum(int arr[],int start,int len)
{ if(start>=len)
   return 0;
  return (arr[start]+sum(arr,start+1,len));
}
```

LEVEL 2

1. **Irfan enjoys**

```
#include <stdio.h>
int i,j;
int minimum(int a,int b)
{ if(a>b)
   return 1;
  else
   return 0;
}
int partition(int arr[],int low,int high)
{ for(i=1;i<=high;i++)
   { if(arr[i]==low)
       { printf("%d\n",i);
         break;
       }
   }
  return 0;
}
void swap(int *a,int *b)
{
    *a=*a + *b;
    *b=*a - *b;
    *a=*a - *b;
}
void quickSort(int arr[],int low,int high)
{ for(i=1;i<=high;i++)
    { for(j=i+1;j<=high;j++)
       { if(minimum(arr[i],arr[j]))
          swap(&arr[i],&arr[j]);
       }
    }
   partition(arr,low,high);
}
int main()
{ int t,n,pos,value,arr[20];
  scanf("%d",&t);
  while(t--)
   { scanf("%d",&n);
     for(i=1;i<=n;i++)
       scanf("%d",&arr[i]);
     scanf("%d",&pos);
    value=arr[pos];
    quickSort(arr,value,n);
   }
  return 0;
}
```

2. Selvan, Aaron

```
#include <stdio.h>
int find1(int arr[], int n){
   int i,pos=-1;
   for(i=0;i<n;i++){
      if(arr[i]==1){
         pos=i;
         break;
         }
   }
   return pos;
}
int main()
{
   int t,n,i,arr[10];
   scanf("%d",&t);
   while(t--){
      scanf("%d",&n);
      for(i=0;i<n;i++)
      scanf("%d",&arr[i]);
      printf("%d\n",find1(arr, n));
   }

         return 0;
}
```

3. Given an Array

```
#include<stdio.h>
char** split_string(char* str)
{
   printf("%c",*str);
   return 0;
}
void plusMinus(int arr_count, int* arr)
{
   float p=0,z=0,n=0;
   int i;
         for (i = 0; i < arr_count; i++)
   {
                  if (arr[i] > 0) {
                           p++;
                  }
                  else if (arr[i] < 0) {
                           n++;
                  }
                  else if (arr[i] == 0) {
                           z++;
                  }
      else if(6<5)
```

```
        {
            split_string("problem");
        }
            }

            printf("%.6f\n",(float)p /arr_count);
            printf("%.6f\n",(float)n /arr_count);
            printf("%.6f\n", (float)z/arr_count);
}
int main()
{
    int arr_count,i;
    scanf("%d",&arr_count);
    int arr[arr_count];
    for(i=0;i<arr_count;i++)
        scanf("%d",&arr[i]);
    plusMinus(arr_count,arr);
    return 0;

}
```

4. Extinct languages

```
#include <stdio.h>
#include<string.h>
void check(char *,int);
char a[100][100],aa[10];
int t,n,k,i;

int main()
{
scanf("%d",&t);
while(t--)
{
    scanf("%d %d",&n,&k);
    for(i=0;i<n;i++)
    scanf("%s", a[i]);
    check(aa,k);
    printf("\n");

}
    return 0;
}

void check(char* w, int k)
{
int z=0,q,j;
char b[100] [100];
while(k--)
{
scanf("%d",&q);
```

```
for(i=0;i<q;i++)
{scanf("%s",b[z]);
z++;
}
}
for(i=0;i<n;i++)
{ int c=0;
for (j=0;j<z;j++)
{if(strcmp(a[i],b[j])==0)
{ c=1;
break;}}

(c>0)?printf("YES "): printf("NO "); }}
```

5. Selvan has

```
#include<stdio.h>
#include<string.h>
int pass(char s[],int n){
   return 0;
}
int main()
{
   int d=0,s=0,l=0,u=0,i,n;
   char str[9999];
   scanf("%i %s",&n,str);
   for(i=0;i<n;i++)
   {
      if(str[i]>='a' && str[i]<='z') l=1;
      else if(str[i]>='A' && str[i]<='Z') u=1;
      else if(str[i]>='0' && str[i]<='9') d=1;
      else if(str[i]=='!'|| str[i]=='@' || str[i]=='#' || str[i]=='$'||
str[i]=='%'|| str[i]=='^'  || str[i]=='&' || str[i]=='*' ||
str[i]=='('|| str[i]==')'  || str[i]=='-' || str[i]=='+')
                                    s=1;
   }
   if(!(n<6))
         printf("%i\n", 4-(s+d+u+l));
  return 0;
}
```

6. Aarav is an

```
#include <stdio.h>
#include<malloc.h>
#include<math.h>
#include<string.h>
#include<stdlib.h>
int *array,counter=0;
void byte_to_binary(int x,int n)
{ static char b[9];
```

```
  b[0]='\0';
  int z;
  for(z=128;z>0;z>>=1)
   strcat(b,((x&z)==z)?"1":"0");
  int i=8-n;
  while(i<8)
   { printf("%c",b[i]);
     i++;
   }
}
void greycode(int n)
{ int k,i;
  array[counter++]=0;
  array[counter++]=1;
  for(i=1;i<n;i++)
   { k=counter-1;
     while(k>=0)
      array[counter++]=array[k--] | 1<<i;
   }
}
int main()
{ int n,i;
  scanf("%d",&n);
  array=(int*)malloc(pow(2,n)*sizeof(int));
  greycode(n);
  for(i=0;i<counter;i++)
   { byte_to_binary(array[i],n);
     printf("\n");
   }

          return 0;
}
```

7. Nancy & Athika

```
#include <stdio.h>
#include<string.h>
void SuperReducedString(char * s,char * u)
{ while(*s!='\0')
   { if(*s==*(s+1))
      s+=2;
    else
     { u=s;
       printf("%c",*u);
       s++;
     }
   }
}
int main()
{ char s[100],u[100];
  scanf("%s",s);
```

```
  SuperReducedString(s,u);

          return 0;
}
```

8. Last week Nathan

```
#include<stdio.h>
#include <string.h>
void patternProcessing(char pattern[]){}
int countFreq();
int main()
{
int t;
scanf("%d",&t);
while(t--)
{
    char txt[100],pat[100];
    scanf("%s%s",txt,pat);
patternProcessing(txt);
printf("%d\n", countFreq(pat, txt));}
return 0;
}
int countFreq(char pat[],char txt[])
{
   int M = strlen(pat),i;
   int N = strlen(txt);
   int res = 0;
   for (i = 0; i <= N- M; i++)
   {
   int j;
   for(j=0; j<M; j++)
      if(txt[i+j]!= pat[j])
         break;
      if (j==M)
      {
      res++;
      j=0;
      }
   }
   if(res==0 || res==1) res=res;
   else if(res==2)res+=1;
   else res+=3;
return res;
}
```

9. Advika & her

```
#include <stdio.h>
#include <string.h>
void insert(long long int hash,long long int position) {}
```

```
int check(long long int hash,long long int position,long long int length) {return 0;}
int main()
{
   char a[1000],b[1000];
   scanf("%s%s", b,a);
   int i,j,x=0; int q; int r;
   for(i=0;i<strlen(a);i++){
      for(j=0;j<strlen(b);j++){
         if(a[i]==b[j]){
            for(q=0; a[i+q]==b[j+q]; q++){q=q;}
            if(q>x){x=q;r=j;}
         }
      }
   }
   for(j=r;j<r+x;j++)
   printf("%c", b[j]);
   printf("\n%d", x);
         return 0;
}
```

10. Caleb found

```
#include <stdio.h>
#include<string.h>
#include<stdlib.h>
int AbsoluteDiff(int a, int b)
{
   return a+b;
}
int count(char str[])
{
  int i,res=0;
  for(i=0;i<strlen(str)/2;i++)
     res+=abs(str[i]-str[strlen(str)-i-1]);
  printf("%d\n",res);
  return 0;
}
int main()
{
   int t;
   scanf("%d",&t);
   while(t--)
   {
      char str[10000];
      scanf("%s",str);
      int a=1,b=0;
      AbsoluteDiff(a,b);
      count(str);
   }
         return 0;
}
```

LEVEL 3

1. The children

```
#include<stdio.h>
#include<limits.h>
#include<malloc.h>
#include<stdlib.h>
#include<math.h>
typedef long long int ll;
ll sum(ll a,ll b){
   return a+b;
}
void buildtree(ll *tree,int *a,int s,int e,int index){
   //single element of array
   if(s==e)
   {
      tree[index]=(ll)a[s];
      return ;
   }

   //base case
   if(s>e)
   return;

   //build left & right subtree take sumof both &put the value in proper index of tree
   int mid=(s+e)/2;

   int lchild=(2*index);
   int rchild=(2*index+1);

   buildtree(tree,a,s,mid,lchild);
   buildtree(tree,a,mid+1,e,rchild);

   ll leftans=tree[lchild];
   ll rightans=tree[rchild];
   tree[index]=leftans+rightans;
}
void updatenode(ll *tree,int index,int s,int e,int i,int value){
   //no overlap
   if(i<s||i>e)
   return;
   if(s==e)
   {
      tree[index]+=(ll)value;
      return;
   }

   int mid=(s+e)/2;
   updatenode(tree,2*index,s,mid,i,value);
```

```
    updatenode(tree,2*index+1,mid+1,e,i,value);

    ll leftans=tree[2*index];
    ll rightans=tree[2*index+1];
    tree[index]=leftans+rightans;
// return ;
}

ll findsum(ll *tree,int index,int qs,int qe,int s,int e){
    //no overlap
    if(qe<s||qs>e)
    return 0;
    if(e<=qe&&s>=qs)
    return tree[index];
    int mid=(s+e)/2;
    ll leftans=findsum(tree,2*index,qs,qe,s,mid);
    ll rightans=findsum(tree,2*index+1,qs,qe,mid+1,e);
    return leftans+rightans;
}
int main()
{
   int n,q,l,r,limit,i,type;
   scanf("%d",&n);
   int a[n];limit=ceil(log(n)/log(2))+1;
   limit=pow(2,limit);
   for(i=0;i<n;i++)
   scanf("%d",&a[i]);
   ll *tree=(ll*)malloc(limit*sizeof(ll));
   int s=0,e=n-1,index=1;
   buildtree(tree,a,s,e,index);
   scanf("%d",&q);
   while(q--)
   {
      scanf("%d %d %d",&type,&l,&r);
       if(type==1)
       {
          ll d;
          ll sum=findsum(tree,index,l-1,r-1,s,e);
          d=(sum/(r-l+1));
          if(sum%(r-l+1)!=0)d++;
          printf("%lld\n",d);
       }
       else
       updatenode(tree,index,s,e,l-1,r);
   }
   return 0;
}
```

2. Mindfire

```
#include<stdio.h>
```

```
#define M 1021
void merge(int a[],int temp[],int low1,int up1,int low2,int up2){
	int i=low1;
	int j=low2;
	int k=low1;
	while((i<=up1)&&(j<=up2)){
		if(a[i]<=a[j])
		temp[k++]=a[i++];
		else
		temp[k++]=a[j++];
	}
	while(i<=up1)
	temp[k++]=a[i++];
	while(j<=up2)
	temp[k++]=a[j++];
	for(i=low1;i<=up2;i++)
	a[i] = temp[i];
}
void sort(int a[],int low,int up){
	int mid;
	int temp[M];
	if(low<up){
		mid=(low+up)/2;
		sort(a,low,mid);
		sort(a,mid+1,up);
		merge(a,temp,low,mid,mid+1,up);
	}
}
int main(){
	int i,n,q,l,r,ans;
	int a[M],b[M];
	scanf("%d",&n);
	for(i=0;i<n;i++){
		scanf("%d",&a[i]);
		b[i] = a[i];
	}
	scanf("%d",&q);
	while(q--){
		ans=0;
		scanf("%d%d",&l,&r);
		sort(b,l-1,r-1);
		for(i=l;i<=r-1;i++){
			ans += (b[i]-b[i-1]) * (b[i]-b[i-1]);
		}
		printf("%d\n",ans);
		for(i=0;i<n;i++)
			b[i] = a[i];
	}
	return 0;
}
```

3. Most problems

```
#include <stdio.h>
#include<stdlib.h>
void inline scanint(int *x);
int main()
{
   int t;
   scanf("%d",&t);
   while(t--)
   {
      int n;
      scanf("%d",&n);
      scanint(&n);
   }
         return 0;
}
void scanint(int *x)
{
   int *ptr,i,s=1;
   ptr =(int*)malloc(*x * sizeof(int));
   for(i=0;i<*x;i++)
      scanf("%d",&ptr[i]);
   int t=ptr[0];
   for(i=1;i<*x;i++)
   if(ptr[i]<=t)
   {
      s=s+1;
      t=ptr[i];
   }
   printf("%d\n",s);
}
```

4. Charan is a young

```
#include <stdio.h>
int partition(int m,int n)
{
   printf("%d",m+n);
   return 0;
}
int small(int a[],int n)
{
   int i, res=1;
   for(i=0;i<n && a[i]<=res;i++)
   {
      res=res+a[i];
   }
   if(6<5)
   {
      partition(1,1);
```

```
    }
    printf("%d\n",res);
    return 0;
}
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int n,i;
        scanf("%d",&n);
        int a[n];
        for(i=0;i<n;i++)
        {
            scanf("%d",&a[i]);
        }
        small(a,n);
    }

            return 0;
}
```

5. The veera Mahendran

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
void exch(int k,int f)
{
    int i;
    for(i=0;i<k;i++){
        printf("%d ",i+k+f);
    }
    for(i=0;i<k;i++){
        printf("%d ",i+f);
    }
    return ;
}

void nextexch(int n,int k,int f)
{
    int ar[n];
    int b[n];
    int pos,count,i;
    for(i=0;i<n;i++)b[i]=i;
    for(i=0;i<k;i++){
        ar[i]=b[i+k];
        b[i+k]=-1;
    }
```

```
    pos=n-k-1;
    count=0;
    while(count<k){
        if(b[pos]!=-1){
            ar[n-1-count]=b[pos];
            b[pos]=-1;
            count++;
        }
        pos--;
    }
    pos=k;
    for(i=0;i<n;i++){
        if(b[i]!=-1){
            ar[pos]=b[i];
            b[i]=-1;
            pos++;
        }
    }
    for(i=0;i<n;i++){
        printf("%d ",ar[i]+f);
    }
    return ;
}

int main() {
    int i,t;
    scanf("%d",&t);
    while(t>0){
        t--;
        int n,k;
        scanf("%d %d",&n,&k);
        if(k==0){
            for(i=0;i<n;i++)printf("%d ",i+1);
            printf("\n");
        }
        else if(n<(2*k))printf("-1\n");
        else{
            int f=1;
            while(n>4*k){
                exch(k,f);
                n=n-2*k;
                f=f+2*k;
            }
            nextexch(n,k,f);
            printf("\n");
        }
    }
    return 0;
}
```

6. Zaikai

```
#include <stdio.h>
#include<stdlib.h>
int cmpfunc(const void *a,const void *b)
{
   return (*(int*)a - *(int*)b);
}
void triplet(int arr[],int N)
{ char c[50]="int partition(int arr[],int low,int high)";
  if(c[0]=='i')
   qsort(arr,N,sizeof(int),cmpfunc);
  int flag=0,i;
  for(i=N-1;i-2>=0;i--)
   { if(arr[i-2]+arr[i-1]>arr[i])
      { flag=1;
        break;
      }
   }
  if(flag)
   printf("YES\n%d %d %d",arr[i],arr[i-1],arr[i-2]);
  else
   printf("NO\n");
}
int main()
{ int n,i;
  scanf("%d",&n);
  int arr[n];
  for(i=0;i<n;i++)
    scanf("%d",&arr[i]);
  triplet(arr,n);
          return 0;
}
```

7. Pankaj lal

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int initcheck();
void del(char a[1000],int i){}

int main()
{
int t;
scanf("%d",&t);
while(t--)
{
char a[1000], b[1000];
scanf("%s %s", a,b);
initcheck(a,b);
```

```
del(a,1);
}
return 0;
}
int initcheck(char a[1000],char b[1000] )
{// char ch[1000];
char *d=a;
d[strlen(d)-1]=0;
int l=strlen(a), c=0,i;
for(i=0;i<l;i++)
if(a[0]==a[1] || a[l-1]==a[l-2])
c++;
if(c>0)
{
if( strcmp(b,d)==0) printf("Yes\n"); else printf("No\n");
}
else printf("No\n"); return 0;
}
```

8. Yasmin

```
#include <stdio.h>
void lazyjem(long long int n,long long int b,long long int m,long long int sum);
void lazyjem(long long int n,long long int b,long long int m,long long int sum)
{ int solved;
  while(n)
   { if(n%2==0)
      solved=n/2;
     else
      solved=(n+1)/2;
    n-=solved;
    sum+=m*solved;
      sum+=b;
   m*=2;
   }
  printf("%lld\n",sum-b);
}
int main()
{ int n,b,m,sum=0,t;
  scanf("%d",&t);
  while(t--)
   { scanf("%d %d %d",&n,&b,&m);
     lazyjem(n,b,m,sum);
   }
          return 0;
}
```

9. Ganga

```
#include <stdio.h>
#include <ctype.h>
```

```
void printInputs(char (*matrix)[100],int R,int C);
int main()
{
        int t,r,c,i;
        scanf("%d",&t);
        while(t--)
        {
           scanf("%d%d",&r,&c);
           char  mat[100][100];
           for(i=0;i<r;i++)
              scanf("%s",mat[i]);
      printInputs(mat,r,c);
        }
        return 0;
}
void printInputs(char (*matrix)[100],int R,int C)
{
   int i,j;
   for(i=0;i<R;i++)
   for(j=0;j<C;j++)
      matrix[i][j]=tolower(matrix[i][j]);
   int count=0;
   for(i=0;i<=R-5&&count==0;i++)
      for(j=0;j<C;j++)

if(matrix[i][j]=='s'&&matrix[i+1][j]=='p'&&matrix[i+2][j]=='o'&&matrix[i+3][j]=='o'&&matrix[i+4][j]=='n
')
           count++;
   for(i=0;i<R;i++)
      for(j=0;j<=C-5&&count==0;j++)

if(matrix[i][j]=='s'&&matrix[i][j+1]=='p'&&matrix[i][j+2]=='o'&&matrix[i][j+3]=='o'&&matrix[i][j+4]=='n
')
           count++;
        if(count>0)
           printf("YES\n");
        else
           printf("NO\n");
}
```

10. Ananthan

```
#include <stdio.h>
const int maxn = 1e7 + 5;
long long int inv[10000005];
void modularInverse(long long int n, long long int prime)
{
   long long int i;
   inv[0] = inv[1] = 1;
   for (i = 2; i <= n; i++)
      inv[i] = inv[prime % i] * (prime - prime / i) % prime;
```

```
}



long long int arrExtended(long long int a,long long int b,long long int *x,long long int *y);
 long long int modInverse(long long int b,long long int m)
{
   long long int x, y;
   long long int g = arrExtended(b, m, &x, &y);
   if (g != 1)
      return -1;
   return (x%m + m) % m;
}
long long int modDivide(long long int a,long long  int b)
{
   long long int m=1000000007;
   long long int inv = modInverse(b, m);
   return (((inv * a) % m)+m)%m;
}
long long int arrExtended(long long int a,long long int b,long long int *x,long long int *y)
{
   if (a == 0)
   {
      *x = 0, *y = 1;
      return b;
   }
   long long int x1, y1;
   long long int arr = arrExtended(b%a, a, &x1, &y1);
   *x = y1 - (b/a) * x1;
   *y = x1;
   return arr;
}
int power(long long int x,long long int y)
{
   long long int res = 1;
   x = x % 1000000007;
   while (y > 0)
   {
      if (y & 1)
         res = (res*x) %1000000007;
      y = y>>1;
      x = (x*x) %1000000007;
   }
   return res%1000000007;
}
long long int modmulti(long long int a,long long int b)
{
   return (a*b)%1000000007;
}
long long int binomialCoeff(long long int n,long long int k)
{
```

```
    long long int res = 1,i;
    for(i = 0; i < k;i++)
    {
        res = modmulti(res,n-i);
        res = modDivide(res,i+1);
    }
    return res;
}
int main()
{

            long long int n,k,a,b,i;
            scanf("%lld %lld %lld %lld",&n,&k,&a,&b);
            long long int res=0;
            if(a==0)
            {
               long long int f=modmulti(b,k);
               res=power(f,n-1);
               res=modmulti(res,k);
               res=res*binomialCoeff(2*(n-1),n-1);
               res=res%1000000007;
               res=modDivide(res,n);
               printf("%lld\n",res);
            }
            else
            {
               modularInverse(maxn - 1 , 1000000007);
               long long int f=modmulti(b,k),p,m=n-1,o=n,q=1;
               p=power(a,m);
               long long int yu=modDivide(1,a);
               res=p;
               for(i=1;i<n;i++)
               {
                 long long int v=modmulti(inv[q],inv[q]);
                 long long int w=modmulti(v,inv[(i+1)]);
                 p=modmulti((yu*p)%1000000007,(i*f)%1000000007);
                 p=(p*(modmulti(o,m)))%1000000007;
                 p=modmulti(p,w);
                 o++;m--;q++;
                 res= (res%1000000007) + (p%1000000007);
               }
               res=modmulti(res,k);
               printf("%lld\n",res);
            }
           return 0;
}
```