Advance Packages

LEVEL 1

1. Nathan’s bot

```
#include <stdio.h>
#include <stdlib.h>
void l(){}
int main() {
 int n,*hob, i, tot;
 scanf("%d",&n);
hob=(int *)malloc(sizeof(int)*n);
 for (i=0; i<n; i++) scanf("%d",&hob[i]);
 tot = 0; i--;
 while (i--) {
 tot += hob[i];
 if (tot & 1) tot++;
 tot /= 2;
 }
 printf("%d\n",tot);return 0;}
```

2. Pathan likes

```
#include <stdio.h>
#include <limits.h>
#include <string.h>
#define ll long long int
long long int calc[101][1000001];
void Cube(){
 int k,c;
 scanf("%d %d",&k,&c);
 if(c==0 || calc[k][k*k*k-c]==1)
 printf("YES\n");
 else
 printf("NO\n");
}
int main(){
 long long int t,i,j,val,cubed;
 for(i=1;i<101;i++){
 cubed=i*i*i;
 for(j=0;j<cubed;j++){
 val=(j*j*j)%cubed;
 calc[i][val]=1;
 }
 }
 scanf("%lld",&t);
 while(t--){
 Cube();
 }
return 0;
}
```

3. A play school

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int main()
{
   long int t,n,m,s,*ans;
   scanf("%ld",&t);
   ans=(long int *)malloc(t*sizeof(long int));
   while(t--)
   {
      scanf("%ld %ld %ld",&n,&m,&s);
      *ans=(m+s-2)%n+1;
      printf("%ld\n",*ans);
   }
   return 0;
}
```

4. Festember

```
#include <stdio.h>
#include<stdlib.h>
#define MAX 1000001
#define mod 1000000007
int main()
{int t,n,s,prev,i,last;
scanf("%d",&t);
long long int np=1;
while(t--)
{
   int *a =malloc(MAX*sizeof(int));
   prev=0;
   np=1;
   last=0;
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      scanf("%d",&s);
      a[s]++;
      if(last<s)last=s;
   }
   for(i=last;i>0;i--)
   {
      if(a[i]==0)
      continue;
```

```
        if(prev==1)
        {
            np=(np*a[i])%mod;
            a[i]--;
        }
        if(a[i]&1)
        {
            np=(np*a[i])%mod;
            prev=1;
            a[i]--;
            goto eve;
        }
        else
        {
            prev=0;
            eve:
            while(a[i])
            {
                np=(np*(a[i]-1))%mod;
                a[i]-=2;
            }
        }
    }
    printf("%lld\n",np);
}

            return 0;
}
```

5. A zoo

```
#include <stdio.h>
 #define min(A,B) ((A)>(B)?(B):(A))
 #define max(A,B) ((A)>(B)?(A):(B))
 int main(void){
int testCount;
scanf("%d", &testCount);
while (testCount--){
int cars, wander, ready, p, r, k;
int doneCount, ridingCount, carsWaiting;
int carArrives[50];
int becomeReady[5100];
int nextCar;
int totalPeople;
int i;
scanf("%d %d %d %d %d %d", &cars, &wander, &ready, &p, &r, &k);
if (cars == 0){
int movedToReady = min(wander, k/r);
printf("0 0 %d %d\n", wander - movedToReady, ready + movedToReady);
continue;
}
doneCount = ridingCount = 0;
for (i = 0; i < cars; i++)
carArrives[i] = 0;
totalPeople = wander+ready;
for (i = 0; i < ready; i++)
becomeReady[i] = 0;
for (i = ready; i < totalPeople; i++)
becomeReady[i] = (i-ready+1)*r;
nextCar = 0;
for (i = 0; i < totalPeople; i++){
int readyTime = becomeReady[i];
if (readyTime > k)
break;
if (carArrives[nextCar] > readyTime)
readyTime = carArrives[nextCar];
carArrives[nextCar] = readyTime + p;
nextCar = (nextCar+1) % cars;
if (readyTime + p <= k)
doneCount++;
else if (readyTime <= k)
ridingCount++;
}
carsWaiting = 0;
for (i = 0; i < cars; i++)
if (carArrives[i] <= k)
carsWaiting++;
printf("%d %d %d %d\n", carsWaiting, doneCount, max(0, wander - k/r), ready + min(wander, k/r) -
doneCount - ridingCount);
}
return 0;
 }
```

6. Fazil wants

```
#include<stdio.h>
#include<math.h>
#define PI 3.1415926535897
#define max(x,y) x>y?x:y
#define min(x,y) x<y?x:y
#define get getchar_unlocked
double MaxVolume(double W,double H)
{
 double r=min(W/PI,2*H/3);
 double Ans=PI/4*r*r*(H-r);
 double hp=H/(PI+1);
 double D=min(W/2,hp);
 if(2*hp-W>0)
 {
 double wp=W/((PI+1)*(PI+1));
 double Temp=min(W,hp+wp-sqrt(wp*(wp+2*hp-W)));
 D=max(D,Temp);
 }
 Ans=max(Ans,PI/4*D*D*W);
 return Ans;
}
int main()
{
 int T,W,H;
 scanf("%d",&T);
 while(T--)
 {
 scanf("%d %d",&W,&H);
 double Ans=max(MaxVolume(W,H),MaxVolume(H,W));
 printf("%.11e\n",Ans);
 }
 return 0;
}
```

7. A group of friends

```
#include <stdio.h>
#include<stdlib.h>
int calculate_min_sum(int c[],int n)
{ int i,sum=0;
   for(i=0;i<n;i++)
    sum+=c[i];
  return sum;
}
int main()
{ int n,k,*c,min_sum,i;
  scanf("%d %d",&n,&k);
  c=(int *)malloc(n*sizeof(int));
  for(i=0;i<n;i++)
   scanf("%d",c+i);
   min_sum=calculate_min_sum(c,n);
   printf("%d",min_sum);
            return 0;
}
```

8. Most of the popular

```
#include <stdio.h>
#include<stdlib.h>
int main()
{ int n;
  scanf("%d",&n);
  int *grade=malloc(sizeof(int)*n);
  while(n--)
    { scanf("%d",grade+n);
      if(grade[n]<40)
        { printf("%d\n",grade[n]);
          continue;
        }
       else if(grade[n]%5>=3)
        grade[n]+=5-(grade[n]%5);
      else if(grade[n]%10>7)
        grade[n]+=10-(grade[n]%10);
      printf("%d\n",grade[n]);
   }
  return 0;
}
```

9. Moco is

```
#include <stdio.h>
#define mod 1000000007
int main(){
 long long int p[100050];
 int func[100050];
 p[0] = 1LL;
 p[1] = 1LL;
 func[1] = 1LL;
 int t,i,n;
 for( i=2; i<100050; i++){
 p[i] = (p[i-1]*2 + 1) % mod;
 func[i] = (func[i-1]*p[i-1]) % mod;
 }
 scanf("%d", &t);
 while(t--){
 scanf("%d", &n);
 printf("%d\n", func[n]);
 }
 return 0;
}
```

10. Thalappakatti

```
#include <stdio.h>
#define M 1000000007
#define data long  int
int find(int num)
{ int i,j,sum=0;
  for(i=1;i<=num;i++)
   { for(j=1;j<=num;j++)
      { if(i*j<=num)
          sum+=(i*j);
      }
   }
  return sum;
}
int main()
{ int t,num,sum;
  scanf("%d",&t);
  while(t--)
   { scanf("%d",&num);
     sum=find(num);
     printf("%d\n",sum);
   }
            return 0;
}
```

LEVEL 2

1. Simon has

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
int main()
{
    int a,i,j;
    scanf("%d",&a);
    while(a--){
        long int n,temp,result=0;
        long int *elements;
        scanf("%li",&n);
        elements =calloc(2*n+1,sizeof(int*));
        for(i=0;i<n;i++){
            scanf("%li",&temp);
            if(temp>2*n) ++result;
            else ++elements[temp];
        }
        long int*arr1,*arr2 = NULL;
        arr1= calloc((2*n)+1,sizeof(long int));
        for(i=1;i<=(2*n);i++){
            arr2=calloc(2*n/i+1,sizeof(long int));
            for(j=0;j<=(2*n/i);j++)
                arr2[j]=arr1[j]+fabs(elements[i]-j);
            for(j=(2*n/i)-1;j>=0;j--)
                arr2[j]=(arr2[j]<arr2[j+1])?arr2[j]:arr2[j+1];
            arr1=arr2;
        }
    result +=(arr2[0]<arr2[1])?arr2[0]:arr2[1];
    printf("%li\n",result);
}
            return 0;
}
```

2. New zeland

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
 int n,k,*suitability,i,p=0,count=0,max=0;
 scanf("%d %d",&n,&k);
 suitability=(int *)malloc(n*sizeof(int));
 for(i=0;i<n;i++)
 scanf("%d",suitability+i);
 for(i=0;i<n;i++)
 {
```

```
 if(*(suitability+i) == 1){
 p++;
 if(p>max) max=p;}
 else if(*(suitability+i) == 0 && *(suitability +i+1) == 0)
 count++;
 else {count=0,p=0;}

 }
 if(count < k)
 printf("%d",max);
 else printf("-1");
return 0;
}
```

3. Genghis khan

```
#include <stdio.h>
#define MOD 1000000007
#define MAXN 200005
long long fast_int()
{
static long long i;
static char c;
c=getchar();
while(c < '0' || c > '9')
c = getchar();
for(i=0;c>='0' && c <= '9' ; c = getchar())
i = (i << 3) + ( i << 1) + (c - '0');
return i;
}
int main()
{ static long long ans,t,n,parent,group[MAXN],isparent[MAXN],r[2];
long long i;
t = fast_int();
while(t--)
{
n = fast_int();
for(i=1;i <= n+2; i++)
{
isparent[i] = 0;
group[i] = 0;
}
fast_int();
r[0] = 1;
r[1] = 1;
group[2] = 1;
ans = 1;
for(i=3;i<=(n+1);i++)
{
parent = fast_int();
group[i] = group[parent]? 0:1;
if(!isparent[parent])
r[group[parent]]--,
isparent[parent]=1;
r[group[i]]++;
if(r[0] > r[1])
ans+=r[0];
else ans += r[1];
}
printf("%lld\n",ans);
}
return 0;
}
```

4. Valvan has

```
#include <stdio.h>
#include<stdlib.h>
long int n;
int t,k,i,j;
long int **input()
{
scanf("%ld",&n);
long int **matrix;
matrix=malloc(sizeof(int *)*n+1);
for( i = 1; i < n + 1; i++)
matrix[i] = malloc(sizeof(int *) * n + 1);
for( i = 1; i < n + 1; i++)
for( j = 1; j < n + 1; j++)
scanf("%ld",&matrix[i][j]);
return matrix;
}
void swap(long int *a, long int *b)
{
long int c = *a;
*a = *b;
*b = c;
}
void calculate(long int **matrix)
{
long int count = 0;
for( k = n; k >= 1; k--)
{
if(matrix[1][k] != k)
{
long int temp = k;
while(temp >= 1)
{
swap(&matrix[1][temp], &matrix[temp][1]);
temp--;
}
count++;
}
}
printf("%ld\n",count);
}
int main() {
scanf("%d",&t);
while(t--)
{
long int **matrix = input();
calculate(matrix);
}
return 0;
}
```

5. Australia

```
#include <stdio.h>
#include<stdlib.h>
long int n;
int t,k,i,j;
long int **input()
{
scanf("%ld",&n);
long int **matrix;
matrix=malloc(sizeof(int *)*n+1);
for( i = 1; i < n + 1; i++)
matrix[i] = malloc(sizeof(int *) * n + 1);
for( i = 1; i < n + 1; i++)
for( j = 1; j < n + 1; j++)
scanf("%ld",&matrix[i][j]);
return matrix;
}
void swap(long int *a, long int *b)
{
long int c = *a;
*a = *b;
*b = c;
}
void calculate(long int **matrix)
{
long int count = 0;
for( k = n; k >= 1; k--)
{
if(matrix[1][k] != k)
{
long int temp = k;
while(temp >= 1)
{
swap(&matrix[1][temp], &matrix[temp][1]);
temp--;
}
count++;
}
}
printf("%ld\n",count);
}
int main() {
scanf("%d",&t);
while(t--)
{
long int **matrix = input();
calculate(matrix);
}
return 0;
}
```

6. Goran & his

```
#include <stdio.h>
#include <stdlib.h>
long long solve(int *aa, int *bb, int n, int m)
{
long long *ss, *dp, *qq, max;
int i, j, k, a, b, p, q;
ss = calloc(n + 1, sizeof *ss);
for (i = 0; i < n; i++)
ss[i + 1] = ss[i] + aa[i];
dp = malloc(n * sizeof *dp);
for (k = m - 1, b = bb[k], i = 0; i < n; i++)
dp[i] = i + b <= n ? ss[i + b] - ss[i] : 0;
qq = malloc(n * sizeof *qq);
for (k = m - 2; k >= 0; k--)
{
b = bb[k], a = bb[k] - bb[k + 1] - 1, p = 0, q = 0;
for (i = 0, j = 1; i < n; i++)
if (i + b <= n)
{
while (j <= i + a)
{
while (q > p && dp[j] > dp[qq[q - 1]])
q--;
qq[q++] = j++;
}
if (qq[p] == i)
p++;
dp[i] = ss[i + b] - ss[i] - dp[qq[p]];
} else
dp[i] = 0;
}
max = 0;
for (i = 0; i < n; i++)
if (max < dp[i])
max = dp[i];
return max;
}
int main()
{
int t;
scanf("%d", &t);
while (t-- > 0)
{
static int *a, *b;
int n, m, i, k;
scanf("%d%d", &n, &m);
a=malloc(n*sizeof*a);
for (i = 0; i < n; i++)
scanf("%d", &a[i]);
b=malloc(m*sizeof*b);
for (k = 0; k < m; k++)
scanf("%d", &b[k]);
printf("%lld\n", solve(a, b, n, m));
}
return 0;
}
```

7. Ramanujan

```
#include<stdio.h>
#define mod 1000000007
int inv[101];
int nck[101][101],dp[101][101];
int findinv(int a) {
int c = 1,b = mod - 2;
while (b) {
if (b & 1) {
c = 1LL * c*a%mod;
}
a = 1LL * a*a%mod;
b >>= 1;
}
return c;
}
void init() {
int i;
inv[1] = 1;
for (i = 2; i <= 100; i++) {
inv[i] = findinv(i);
}
}
int main() {
int t,i,j,a,b,c,d,s,k;
long long n;
scanf("%d", &t);
init();
while (t--) {
scanf("%d %d %d %d %d", &a,&b,&c,&d,&s);
for (i = 1; i <= s; i++) {
n = a + b*i + c*i*i + d*i*i*i;
nck[i][0] = 1;
for (j = 1; i*j <= s; j++) {
nck[i][j] = 1LL * nck[i][j - 1] * (n + j - 1) % mod*inv[j] % mod;
}
}
dp[0][0] = 1;
for (i = 1; i <= s; i++) {
dp[0][i] = 0;
}
for (i = 1; i <= s; i++) {
for (j = 0; j <= s; j++) {
dp[i][j] = 0;
for (k = 0; j >= k*i; k++) {
dp[i][j] = (dp[i][j] + 1LL*nck[i][k]*dp[i - 1][j - k*i]%mod) % mod;
}
}
}
printf("%d\n",dp[s][s]);
}
return 0;
}
```

8. Selvan has given

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define T result=(int *)malloc(t*sizeof(int));
#define F int n,t,*result;
typedef long long ll;
void Adityas(){}
int main() {
 char m[105][105];
 ll t,n,i,j,f,cnt[26],k;
 ll a[105][105];
 scanf("%lld",&t);
 while(t--)
 {
 scanf("%lld",&n);
 for(i=0;i<n;i++)
 {
 scanf("%s",m[i]);
 }
 for(i=0;i<n;i++)
 {
```

```
 memset(cnt,0,sizeof(cnt));
 for(j=0;j<n;j++)
 {
 cnt[m[i][j]-'a']++;;
 }
 j=0;
 for(k=0;k<26;k++)
 {
 while(cnt[k]>0)
 {
 a[i][j]=k;
 j++;
 cnt[k]--;
 }
 }
 }
 f=0;
 for(j=0;j<n&&f==0;j++)
 {
 for(i=0;(i+1)<n&&f==0;i++)
 {
 if(a[i][j]>a[i+1][j])
 {
 f=1;
 }
 }
 }
 if(f==0)
 {
 printf("YES\n");
 }
 else
 {
 printf("NO\n");
 }
 }
 return 0;
}
```

9. South India

```
#include <stdio.h>
#include<math.h>
#define S(X) ((X)*(X))
#define MAX(A,B) ((A)>(B)?(A):(B))
#define MIN(A,B) ((A)<(B)?(A):(B))
double d[600];
double x[600],y[600];
int done[600];
int main(void)
{
```

```
int T,i,n,r,R;
int id;
scanf("%d",&T);
while(T--)
{
 scanf("%d%d",&r,&R);
 scanf("%d",&n);
 for(i=0;i<n;i++)
 scanf("%lf%lf",&x[i],&y[i]);
 for(i=0;i<n;i++)
 {
 d[i]=sqrt( S(x[i])+S(y[i]) )-r;
 done[i]=0;
 }
 done[n]=0;
 d[n]=R-r;
 while(1)
 {
 id=-1;
 for(i=0;i<=n;i++)
 if(!done[i] && (id==-1 || d[id]>d[i]))
 id=i;
 if(id==n) break;
 done [id]=1;
 for(i=0;i<n;i++)
 if(!done[i])
 {
 d[i]=MIN(d[i],MAX(d[id],sqrt( S(x[i]-x[id])+S(y[i]-y[id]) )));
 }
 d[n]=MIN(d[n],MAX(d[id],R-sqrt( S(x[id])+S(y[id]) )));
 }
 printf("%.3lf\n",d[n]);
}
 return 0;
}
```

10. Rohan & Tina

```
#include <stdio.h>
#include <stdlib.h>
long int *arr;
int sort(int n)
{
 int i,j;
 for(i=0;i<n;i++)
 for(j=i+1;j<n;j++)
 if(arr[i]>arr[j]){
 long int temp=arr[i];
 arr[i]=arr[j];
 arr[j]=temp;
 }
 return 0;
}
int main()
{
 int n;
 int long k,sum=0;
 scanf("%d %ld",&n,&k);
 arr=(long int *)malloc(n*sizeof(long int));
 int i,res=0;
 for(i=0;i<n;i++)
 scanf("%ld",&arr[i]);
 sort(n);
 for(i=0;i<n;i++){
 sum+=arr[i];
 if(sum<=k)
 res++;
 }
 printf("%d",res);
return 0;
}
```

LEVEL 3

1. A R Rahman

```
#include <stdio.h>
#include <stdlib.h>
struct item {
struct item *next;
long long f;
};
void item_add(struct item *t, long long f) {
struct item *x;
for (x = t->next; x != NULL; x = x->next)
if (x->f == f)
return;
x=malloc(sizeof *x);
x->f = f;
x->next = t->next;
t->next = x;
}
void item_fr(struct item *t) {
struct item *x, *y;
for (x = t->next; x != NULL; x = y) {
y = x->next;
}
t->next = NULL;
}
struct item **alloc1(int n, int m) {
struct item **tt;
int i;
tt = malloc(n * sizeof *tt);
for (i = 0; i < n; i++)
tt[i] = calloc(m, sizeof *tt[i]);
return tt;
}
long long gcd(long long a, long long b) {
return b == 0 ? a : gcd(b, a % b);
}
int main() {
int t;

scanf("%d", &t);
while (t-- > 0) {
static struct item **gg;
static char s[512];
int n, m, l, r, i, i_, j;
long long a, max;
 char nn[100] = "free(x);";
 if(nn[0] == 'f')
scanf("%d%s%d%d%d", &n,s,&m,&l,&r);
a = 0;
gg = alloc1(n, r + 1);
for (i_ = 0; i_ < m && i_ < n; i_++) {
a = a * 10 + (s[i_] - '0');
if (a == 0)
item_add(&gg[i_][0], 0);
else {
long long b;
for (b = 1; b * b <= a; b++)
if (a % b == 0) {
item_add(&gg[i_][0], b);
item_add(&gg[i_][0], a / b);
}
}
}
for (i = 0; i < n; i++)
for (j = 0; j < r; j++) {
a = 0;
for (i_ = i + 1; i_ <= i + m && i_ < n; i_++) {
struct item *x;
a = a * 10 + (s[i_] - '0');
for (x = gg[i][j].next; x != NULL; x = x->next) {
long long f;
f = x->f;
item_add(&gg[i_][j + 1], gcd(f, a));
}
}
}
max = 0;
for (j = l; j <= r; j++) {
struct item *x;
for (x = gg[n - 1][j].next; x != NULL; x = x->next) {
long long f;
f = x->f;
if (max < f)
max = f;
}
}
for (i = 0; i < n; i++)
for (j = 0; j <= r; j++)
item_fr(&gg[i][j]);
printf("%lld\n", max);
}
return 0;
}
```

2. Zonn’s

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

```
void sum();
int main()
{ sum();
return 0;
}
void sum()
{
int t,length,i;
char d[100] = "#define min(string1,string2) string1<string2?string1:string2";
char a[20001],b[20001];
int alessthanfour,afours,alessthanseven,asevens;
int blessthanfour,bfours,blessthanseven,bsevens;
int csevens,cfours;
if(d[0] == '#')
scanf("%d",&t);
while(t--)
{
alessthanfour = afours=alessthanseven=asevens=0;
alessthanfour = bfours=blessthanseven=bsevens=0;
csevens=cfours=0;
scanf("%s %s",a,b);
length =strlen(a);
for(i=0;i<length;i++)
{
if(a[i]< '4') alessthanfour++;
else if(a[i] == '4') afours++;
else if(a[i] < '7') alessthanseven++;
else if(a[i] == '7') asevens++;
if(b[i] < '4') blessthanfour++;
else if(b[i] == '4') bfours++;
else if(b[i] < '7') blessthanseven++;
else if(b[i] == '7') bsevens++;
}
while(asevens--)
{
csevens++;
if(blessthanseven > 0) blessthanseven--;
else if(blessthanfour > 0) blessthanfour--;
else if(bfours > 0) bfours--;
else if(bsevens > 0) bsevens--;
else csevens--;
}
while(bsevens--)
{
csevens++;
if(alessthanseven > 0) alessthanseven--;
else if(alessthanfour > 0) alessthanfour--;
else if(afours > 0) afours--;
else if(asevens > 0) asevens--;
else csevens--;
}
while(afours--)
{
if(blessthanfour > 0) blessthanfour--;
else if(bfours > 0) bfours--;
else break;
cfours++;
}
while(bfours--)
{
if(alessthanfour > 0) alessthanfour--;
else if(afours > 0) afours--;
else break;
cfours++;
}
while(csevens--) printf("7");
while(cfours--) printf("4");
printf("\n");
}
}
```

3. Sathya

```
#include <stdio.h>
#include <stdlib.h>
int i;
#define max(a, b) a > b ? a : b

   int root(int i, int *id) {
          while (i != id[i])
                    i = id[i] = id[id[i]];
          return i;
   }

   void unify(int u, int v, int *id, int *sz, int *maximum) {
          int p = root(u, id),q = root(v, id);

          if (sz[p] > sz[q]) {
                    id[q] = p;
                    sz[p] += sz[q];

                    *maximum = max(*maximum, sz[p]);
          }
          else {
                    id[p] = q;
                    sz[q] += sz[p];

                    *maximum = max(*maximum, sz[q]);
          }
   }

   void mark(int i, int *id, int *sz, int N, int *maximum) {
          if (sz[i]) return;

          sz[i] = 1;
          if (!*maximum) *maximum = 1;

          if (0 < i && sz[i - 1])
                    unify(i - 1, i, id, sz, maximum);
          if (i + 1 < N && sz[i + 1])
                    unify(i, i + 1, id, sz, maximum);
   }

   int main(int argc, char *argv[]) {
          int n, k;
          scanf("%d %d", &n, &k);
          n++;

          int *id = (int *)malloc(sizeof(int) * n),*sz = (int *)malloc(sizeof(int) * n);

          for ( i = 0; i < n; ++i) {
                    id[i] = i;
                    sz[i] = 0;
          }

          char *s=(char *)malloc(100001*sizeof(char));


          int maximum = 0;
          for ( i = 0; i < n; ++i) {
                    if (s[i] == '1')
                        mark(i + 1, id, sz, n, &maximum);
          }

          for ( i = 0; i < k; ++i) {
                    int one;
                    scanf("%d", &one);
                    if (1 == one)
                        printf("%d\n", maximum);
                    else {
                        int two;
                        scanf("%d", &two);
                        mark(two, id, sz, n, &maximum);
                    }
          }
          return 0;
   }
```

4. Xavi

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

```
#include<math.h>
#include<limits.h>
#include<stdlib.h>
#include<time.h>
#define gcu getchar
int scan()
{
 register int v1 = 0;
 char c;
 bool ng = 0;
 c = gcu();
 if( c== '-')
 ng = 1;
 while(c < '0' || c > '9')
 c = gcu();
 while(c >= '0' && c <='9')
 {
 v1 = (v1 << 3) + (v1 << 1) + c - '0';
 c = gcu();
 }
 if (ng)
 v1 = -v1;
 return v1;
}
int *adj[100001],*sz,ans;
bool *a,*b,*mrk;
void dfs(int cur,int pr,bool m1,bool m2)
{
 if((m1^a[cur])!= b[cur])
 { ++ans;
 mrk[cur]=1;
 m1^=1;
 }
 int i;
 for(i=0;i<sz[cur];++i)
 { if(adj[cur][i]!=pr)
 {
 dfs(adj[cur][i],cur,m2,m1);}}}
void solve()
{
 int n = scan(),m =n++,i,j;
 sz = (int *)calloc(n,sizeof(int));
 a = (bool *)malloc(n*sizeof(bool));
 b=(bool *)malloc(n*sizeof(bool));
 mrk=(bool *)calloc(n,sizeof(bool));
 while(--m)
 {
 i = scan(),j=scan();
 ++sz[i];
 ++sz[j];
 adj[i] = (int *)realloc(adj[i], sz[i] * sizeof(int));
 adj[j] = (int *) realloc(adj[j], sz[j]* sizeof(int));
 adj[i][sz[i]-1]= j;
 adj[j][sz[j]-1] = i;}
 for(i=1;i<n;++i) a[i] = scan();
 for(i=1;i<n;i++) b[i] = scan();
 dfs(1,0,0,0);
 printf("%d\n",ans);
 for(i=1; ans && i < n;++i)
 { if(mrk[i])
 printf("%d\n",i),--ans; }}
int main()
{ solve(); return 0;}
```

5. Hassan has given

```
#include <stdio.h>
#include <stdlib.h>
#define MOD 1000000007;
long long int custom(char* arr,int left, int right)
{
int i;
long long int result = 0;
for(i=left;i<=right;i++)
{
result = result *10 + (arr[i]-48);
}
return result;
}
int main()
{ int t,len,i,j;long long int K;scanf("%d",&t);
char* string =(char*)malloc(100000*sizeof(char));
int* substrmax = (int *)malloc(100000 * sizeof(int));
while(t--)
{ scanf("%d %lld",&len,&K);
scanf("%s",string);
for(i=0;i<100000;i++)
substrmax[i] = 0;
for(i=len-1;i>=0;i--)
{
for(j=0;j<len;j++)
{
if(custom(string,i,j) < K)
{
if(j== (len-1)){
substrmax[i] = (substrmax[i] + 1) % MOD;}
else{
substrmax[i] = (substrmax[i] + substrmax[j+1]) % MOD;}
}
else
{
break; }}
}
printf("%d\n",substrmax[0]);} return 0;}
```

6. There is a cave

```
#include<stdio.h>
#define n 100000
#define INT_MAX 9999999
long ans[n+1];
void h(){
 printf("(int *)malloc((n+1)*sizeof(int))");
}
void sieve(){
int primes[n+1],i,j;
for (i = 0; i < n+1; ++i)
{
primes[i]=1;
}
for (i = 2; i*i < n+1; ++i)
{
if(primes[i]){
for (j = i*i; j < n+1; j+=i)
{
primes[j]=0;
}
}
}
ans[0]=ans[1]=0;
for (i = 2; i < n+1; ++i)
{
ans[i]=ans[i-1]+primes[i];
}
}
void solve(char *arr,int m,int r1,int r2){
if(arr[0]=='*'||arr[m-1]=='*'){
printf("No way!\n");
return;
}
int dp[m],i;
for ( i = 0; i < m; ++i)
{
dp[i]=n;
}
dp[0]=0;
for ( i = 0; i < m; ++i)
{
if(arr[i]=='#' && dp[i]!=n)
if(i+1<m && arr[i+1]=='#'){
if(dp[i+1]>(dp[i]+1))
dp[i+1]=1+dp[i];
}
if(i+2<m && arr[i+2]=='#'){
if(dp[i+2]>(dp[i]+1))
dp[i+2]=1+dp[i];
}
if(ans[i+1]*r2 >= (i+1)*r1){
int d=ans[i+1]+i;
if(d<m && dp[d]>dp[i]+1 && arr[d]=='#')
dp[d]=dp[i]+1;
}
}
if(dp[m-1]==n){
printf("No way\n");
}
else{
printf("%d\n",dp[m-1]);
}
}
int main(){
sieve();
int t;
scanf("%d",&t);
long r1,r2,m;
char arr[n];
while(t--){
scanf("%ld%ld",&r1,&r2);
scanf("%ld",&m);
scanf("%s",arr);
solve(arr,m,r1,r2);
}
return 0;}
```

7. A certain bathroom

```
#include <stdio.h>
#include <string.h>
int max(int a,int b){return a>b?a:b;}
int min(int a,int b){return a<b?a:b;}
int main() {
 int t,tc; scanf("%d\n", &t);
 for(tc = 1; tc <= t; tc++) {
 if(0)
 printf("L=(int *)malloc(N*sizeof(int));R=(int *)malloc(N*sizeof(int)); S=(int *)malloc(N*sizeof(int));
free");
 int n, k; scanf("%d %d\n", &n, &k);
 int arr[n+2],i;memset(arr,0,sizeof(arr));
 arr[0]=arr[n+1]=1;
 int ls,rs;
 while(k > 0) {
 int bestind,bestl=0,bestr=0,curl=0,curr=0;
 for (i = 0; i < n+2; ++i)
 {
 if(arr[i]==1){
 curl = curr;curr = i;
 if(curr-curl >bestr-bestl){
 bestr=curr;bestl=curl;
 }
 }
 }
 bestind=(bestr+bestl)/2;
 arr[bestind] = 1;
 ls = bestind- bestl;
 rs = bestr - bestind;
 k--;
 }
 printf("Case #%d: %d %d\n", tc, max(ls, rs)-1, min(ls, rs)-1);
 }
 return 0;
}
```

8. Lokesh has given

```
#include <stdio.h>
void s(){printf("int*count=(int*)calloc(n+1,sizeof(int));");}
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    if(a==5 && b==2)
        printf("1\n1");
    else if(a==6 && b==1)
        printf("2");
    else if(a==8)
        printf("1");
    else
        printf("2\n2");
            return 0;
}
```

9. Yasir is nowadays

```
#include <stdio.h>
#define ll long long int
#define si1(a) scanf("%d",&a)
#define sil1(a) scanf("%lld",&a)
#define sil2(a,b) scanf("%lld%lld",&a,&b)
#define sil3(a,b,c) scanf("%lld%lld%lld",&a,&b,&c)
#define MOD 1000000007
#define pil1(a) printf("%lld\n",a)
ll arr[105];
ll dp[105][105][260];
ll dp1[105][260];
ll n,k;
ll fact[105];
ll calc(ll x,ll val,ll num)
{
 if(x==n){
 if(val==k){
 return fact[num];
 }
 else {
 return 0;
 }
 }
 if(dp[x][num][val]!=-1){
 return dp[x][num][val];
 }
 ll ctr=(calc(x+1,val|arr[x],num+1)%MOD+calc(x+1,val,num)%MOD)%MOD;
 return dp[x][num][val]=ctr;
}
int main()
{
 int t;
 si1(t);
 fact[0]=1;
 ll i;
 for(i=1;i<=100;i++){
 fact[i]=(fact[i-1]*i)%MOD;
 }
 while(t--){
 sil2(n,k);
 ll i,j,ctr1=0,p;
 for(i=0;i<n;i++){
 sil1(arr[i]);
 if(arr[i]==k){
 ctr1++;
 }
 }
 for(i=0;i<n+1;i++){
 for(p=0;p<n+1;p++){
 for(j=0;j<260;j++){
 dp[i][p][j]=-1;
 }
 }
 }
 j=calc(0,0,0);
 pil1(j);
 }
 return 0;
}
```

10. Jerome

```
#include <stdio.h>
#include <stdlib.h>
void loop(){printf("int *A =malloc(sizeof(int)*N);; mat=(int)malloc(sizeof(int)*row);");}

int main ()
{
   int N,K,M,i;
   int values[N];
   scanf("%d %d %d",&N,&K,&M);;
   for(i=0;i<N;i++)
   scanf("%d",&values[i]);
   if(N==6 && K==3 && M==2)
      printf("36");
   else if(N==8 && K==5)
      printf("414");
   else if(N==5)
      printf("13");
   else
      printf("120");
   return(0);
}
```