A. Captain Flint and Crew Recruitment

只在前四个里挑三个就可以，注意不可以有相同的。

```
#include<bits/stdc++.h>
#define ll long long
#define fi first
#define se second
#define mp make_pair
#define pb push_back
#define pii pair<int,int>
using namespace std;
const int N=1e5+10;
const ll inf=0x3f3f3f3f;
int num[4]={6,10,14,15};
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int n,f=0,a,b,c,d;
        scanf("%d",&n);
        for(int i=0;i<4;i++)
        for(int j=i+1;j<4;j++)
        for(int k=j+1;k<4;k++)
        {
            int t=n-num[i]-num[j]-num[k];
            if(t==num[i]||t==num[j]||t==num[k]||t<=0)continue;
            f=1,a=num[i],b=num[j],c=num[k],d=t;
        }
        if(!f)puts("NO");
        else printf("YES\n%d %d %d %d\n",a,b,c,d);

    }
    return 0;
}
```

B. Captain Flint and a Long Voyage

因为要占尽可能多的位必定会是 $99\ldots988\ldots 8$ 的形式，把后面多了 1 也会被去掉的部分变成 8 即可。

```
#include<bits/stdc++.h>
#define ll long long
#define fi first
#define se second
#define mp make_pair
#define pb push_back
#define pii pair<int,int>
using namespace std;
const int N=1e5+10;
const ll inf=0x3f3f3f3f;
int a[N];
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int n;
        scanf("%d",&n);
        for(int i=1;i<=n;i++)a[i]=9;
        int j=n,k=1;
        for(int i=1;i<=n;i++)
        {
            if(k!=8)a[j]|=k,a[j]^=k;
            k<<=1;if(k>8)k=1,j--;
        }
        for(int i=1;i<=n;i++)
        printf("%d",a[i]);
        puts("");
    }
    return 0;
}
```

C. Uncle Bogdan and Country Happiness

一棵树，每个节点 p_i 个人，大家在 1 工作，下班之后按最短路回家，心情只能变差，每个城市的幸福值是途径该城市好心情的人减去坏心情的人，问给出的幸福值是否可行。

可以以幸福值和途径该城市的人数直接算出每个城市好心情的人和坏心情的人，判定 $父亲好心情\ge儿子好心情+\max(0,父亲城市住的人-父亲坏心情)$

```
#include<bits/stdc++.h>
#define ll long long
#define fi first
```

```
#define se second
#define mp make_pair
#define pb push_back
#define pii pair<int,int>
using namespace std;
const int N=1e5+10;
const ll inf=0x3f3f3f3f;
int n,m,head[N],cnt;
ll p[N],h[N],x[N],y[N],sz[N];
struct Node{int nxt,to;}edges[N*2];
void addedge(int u,int v){edges[++cnt]=Node{head[u],v},head[u]=cnt;}
bool dfs(int u,int f)
{
    sz[u]=p[u];
    for(int i=head[u];~i;i=edges[i].nxt)
    {
        int v=edges[i].to;
        if(v!=f)
        {
            if(!dfs(v,u))return 0;
            sz[u]+=sz[v];
        }
    }
    if((h[u]+sz[u])&1)return 0;
    x[u]=(h[u]+sz[u])/2,y[u]=x[u]-h[u];
    if(x[u]<0||y[u]<0)return 0;
    ll sonx=0;
    for(int i=head[u];~i;i=edges[i].nxt)
    {
        int v=edges[i].to;
        if(v!=f)sonx+=x[v];
    }
    return x[u]>=sonx+max(0LL,p[u]-y[u]);
}
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        cnt=0;
        memset(head,-1,sizeof(head));
        scanf("%d%d",&n,&m);
        for(int i=1;i<=n;i++)scanf("%lld",&p[i]);
        for(int i=1;i<=n;i++)scanf("%lld",&h[i]);
        for(int i=1;i<n;i++)
        {
            int x,y;
            scanf("%d%d",&x,&y);
            addedge(x,y),addedge(y,x);
        }
```

```
        if(dfs(1,0))puts("YES");
        else puts("NO");

    }
    return 0;
}
```

D. Captain Flint and Treasure

给长度 n 的数组 a,b 每次选一个位置给答案加上 a_i ，然后给 a_{b_i} 加上 a_i （如果不为 -1）。

因为条件保证没有环，给 b_i 向 i 连边形成的是一个森林，dfs去处理，如果某个儿子的最终 a_v 大于零就把它放在当前节点前面，并给当前节点的 a_u 加上该值，否则放在后面。

```
#include<bits/stdc++.h>
#define ll long long
#define fi first
#define se second
#define mp make_pair
#define pb push_back
#define pii pair<int,int>
using namespace std;
const int N=2e5+10;
const ll inf=0x3f3f3f3f;
int head[N],cnt;
struct Node{int nxt,to;}edges[N];
void addedge(int u,int v){edges[++cnt]=Node{head[u],v},head[u]=cnt;}
ll a[N],b[N],num[N],res;
void dfs(int u)
{
    num[u]=a[u];
    for(int i=head[u];~i;i=edges[i].nxt)
    {
        int v=edges[i].to;dfs(v);
        if(num[v]>0)num[u]+=num[v];
    }
    res+=num[u];
}
void print(int u)
{
    for(int i=head[u];~i;i=edges[i].nxt)
    {
        int v=edges[i].to;
        if(num[v]>0)print(v);
```

```
    }
    printf("%d ",u);
    for(int i=head[u];~i;i=edges[i].nxt)
    {
        int v=edges[i].to;
        if(num[v]<=0)print(v);
    }
}
int main()
{
    memset(head,-1,sizeof(head));
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++)scanf("%lld",&a[i]);
    for(int i=1;i<=n;i++)
    {
        scanf("%lld",&b[i]);
        if(b[i]!=-1)addedge(b[i],i);
    }
    for(int i=1;i<=n;i++)if(b[i]==-1)dfs(i);
    printf("%lld\n",res);
    for(int i=1;i<=n;i++)if(b[i]==-1)print(i);
    puts("");
    return 0;
}
```