#include<bits/stdc++.h>

#define llINF 9223372036854775807

#define pi 3.141592653589793//23846264338327950254

#define endl "\n"

#define mp make\_pair

#define pb push\_back

#define ll long long

#define int long long

#define IOS ios::sync\_with\_stdio(0); cin.tie(0); cout.tie(0);

**using** **namespace** std;

**const** ll maxn=1e3+7;

**const** **double** eps=1e-10;

**const** ll mod=1e9+7;

ll n;

ll totA=0,totB=0;

ll casA[maxn\*2],casB[maxn\*2];   //cas[i]记录A和B两个人第i次胜利出现在第几个数据

ll tot=0;

**struct** Node

{

    ll round,score;

};

Node node[maxn];

**bool** cmp(Node a,Node b)

{

**if**(a.round<b.round) **return** 1;

**if**(a.round==b.round&&a.score<b.score) **return** 1;

**return** 0;

}

int32\_t main()

{

    IOS;

    cin>>n;

**for**(ll i=1;i<2\*maxn;i++) casA[i]=casB[i]=llINF; //注意这里开两倍的数组以及初始化值方便后续比较，

**bool** flagA=0,flagB=0;                           //记录最后一次胜利是A还是B，这个决定了谁才是胜出者

**for**(ll i=1;i<=n;i++)

    {

        ll x;

        cin>>x;

**if**(x==1) casA[++totA]=i;

**else** casB[++totB]=i;

**if**(i==n)

        {

**if**(x==1) flagA=1;

**else** flagB=1;

        }

    }

**for**(ll i=n;i>=1;i--)//这里虽然暴力for了一遍n，但是里面循环部分的步长为i，也就是说总共的复杂度实际上是n/1+n/2+n/3+....+n/n=nlogn

    {

        ll winA=0,winB=0,tarA=0,tarB=0;//win为获胜轮数，tarA为当前A已经胜利了几局

        ll now=1;   //now记录当前已经处理数据的下一个位置，决定是否跳出循环

**bool** flag=1;    //标记本轮是否成功找到合适的局数

**while**(now<=n)

        {

**if**(casA[tarA+i]==llINF&&casB[tarB+i]==llINF)    //如果A和B下一个获胜的位置均已经超出了n，这里的判断方便得益于之前的两倍数组和预处理

            {

                flag=0;

**break**;

            }

**if**(casA[tarA+i]<casB[tarB+i])   //tarA为已经胜利的局数，那么他如果要再赢一轮的话，局数应该是tarA+i，比较两者当中下标小的那一个为本轮胜利者

            {

                winA++;

                tarA+=i;

                tarB=casA[tarA]-tarA;   //casA[tarA]代表A胜利tarA局的数据下标位置，也就是当前进行的总局数，减去A胜利的局数也就是B胜利的局数了

                now=casA[tarA]+1;

            }

**else**

            {

                winB++;

                tarB+=i;

                tarA=casB[tarB]-tarB;

                now=casB[tarB]+1;

            }

        }

**if**(winA==winB) flag=0;  //特判平局的情况

**if**(winA>winB&&flagB) flag=0;    //特判A胜利但是最后一局获胜者是B的情况

**if**(winA<winB&&flagA) flag=0;    //特判B胜利但是最后一局获胜者是A的情况

**if**(flag)

        {

            node[++tot].round=max(winA,winB);

            node[tot].score=i;

        }

    }

    sort(node,node+tot,cmp);

    cout<<tot<<endl;

**if**(tot)

    {

**for**(ll i=1;i<=tot;i++) cout<<node[i].round<<' '<<node[i].score<<endl;

    }

}