1 pallindrome

```
def is_pallindrome(s):
  n = len(s)
  if n==1:
    return True
  for i in range(n//2):
    if s[i] != s[-i-1]:
      return False
  return True
s = input()
def f(s):
  for a in range(1,len(s)-2):
    if is_pallindrome(s[:a]):
      for b in range(a+1,len(s)):
        if is_pallindrome(s[a:b]) and is_pallindrome(s[b:]):
          print(s[:a])
          print(s[a:b])
          print(s[b:])
          return
  print("not possible")

f(s)
```

```
#2string compares
table = {0: 'zero', 1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five', 6: 'six', 7: 'seven', 8: 'eight', 9: 'nine',
10: 'ten', 11: 'eleven', 12: 'twelve', 13: 'thirteen', 14: 'fourteen', 15: 'fifteen', 16: 'sixteen', 17:
'seventeen', 18: 'eighteen', 19: 'nineteen', 20: 'twenty', 21: 'twentyone', 22: 'twentytwo', 23:
'twentythree', 24: 'twentyfour', 25: 'twentyfive', 26: 'twentysix', 27: 'twentyseven', 28: 'twentyeight',
29: 'twentynine', 30: 'thirty', 31: 'thirtyone', 32: 'thirtytwo', 33: 'thirtythree', 34: 'thirtyfour', 35:
'thirtyfive', 36: 'thirtysix', 37: 'thirtyseven', 38: 'thirtyeight', 39: 'thirtynine', 40: 'forty', 41: 'fortyone',
42: 'fortytwo', 43: 'fortythree', 44: 'fortyfour', 45: 'fortyfive', 46: 'fortysix', 47: 'fortyseven', 48:
'fortyeight', 49: 'fortynine', 50: 'fifty', 51: 'fiftyone', 52: 'fiftytwo', 53: 'fiftythree', 54: 'fiftyfour', 55:
'fiftyfive', 56: 'fiftysix', 57: 'fiftyseven', 58: 'fiftyeight', 59: 'fiftynine', 60: 'sixty', 61: 'sixtyone', 62:
'sixtytwo', 63: 'sixtythree', 64: 'sixtyfour', 65: 'sixtyfive', 66: 'sixtysix', 67: 'sixtyseven', 68: 'sixtyeight',
69: 'sixtynine', 70: 'seventy', 71: 'seventyone', 72: 'seventytwo', 73: 'seventythree', 74: 'seventyfour',
75: 'seventyfive', 76: 'seventysix', 77: 'seventyseven', 78: 'seventyeight', 79: 'seventynine', 80:
'eighty', 81: 'eightyone', 82: 'eightytwo', 83: 'eightythree', 84: 'eightyfour', 85: 'eightyfive', 86:
'eightysix', 87: 'eightyseven', 88: 'eightyeight', 89: 'eightynine', 90: 'ninety', 91: 'ninetyone', 92:
'ninetytwo', 93: 'ninetythree', 94: 'ninetyfour', 95: 'ninetyfive', 96: 'ninetysix', 97: 'ninetyseven', 98:
'ninetyeight', 99: 'ninetynine', 100: 'hundred'}
vowels = {'a', 'e', 'i', 'o', 'u'}

n = int(input())
ls = list(map(int, input().split()))

d = 0

def wordify(x) -> int:
    if x < 0 or x > 100:
        return

    su = 0
    for c in table[x]:
        if c in vowels:
            su += 1
    return su

def pair_sum(d, ls):
    res = []
    while ls:
        num = ls.pop()
        diff = d - num
        if diff in ls:
            res.append([diff, num])
    res.reverse()
    return res
```

```
for i in ls:
    d += wordify(i)


# print(d)
print(table[len(pair_sum(d, ls))])
```

```
#3Corona
#include<iostream>
#include<string>
#include<bits/stdc++.h>
using namespace std;


int main()
{
string fever,cough,fatigue,sneezing,aches,nose,throat,diarea,headache,breath; //declaring the variables
cout << "\nCheck for whether you have coronavirus,cold or flu or none of them.: ";
    //asking the user to input the symptoms
cout << "\nDo you have fever:(common/rare/no): ";
cin >> fever;


cout << "Do you have fatigue:(common/sometimes/NO): ";
cin >> fatigue;
cout << "Do you have cough:(common/mild/NO): ";
cin >> cough;
cout << "Do you have sneezing:(common/NO): ";
cin >> sneezing;
cout << "Do you have aches:(common/sometimes/NO): ";
cin >> aches;
cout << "Do you have Runny or stuffy nose:(common/rare/sometimes/NO): ";
cin >> nose;
cout << "Do you have Sore throat:(common/sometimes/NO): ";
cin >> throat;
cout << "Do you have diarrhea:(rare/sometimes/NO): ";
cin >> diarea;
cout << "Do you have headaches:(common/rare/sometimes/NO): ";
cin >> headache;
cout << "Do you have Shortness of breath:(sometimes/NO): ";
cin >> breath;
  if(fever== "COMMON" && fatigue=="SOMETIMES" && cough== "COMMON" && sneezing == "NO" && aches== "SOMETIMES" && nose=="RARE" && throat=="SOMETIMES" && diarea=="RARE" && headache=="SOMETIMES" && breath=="SOMETIMES")
{
cout << "You have coronavirus!!!";
}
else if(fever== "RARE" && fatigue=="SOMETIMES" && cough== "MILD" && sneezing == "COMMON" && aches== "COMMON" && nose=="COMMON" && throat=="COMMON" && diarea=="NO" && headache=="RARE" && breath=="NO")
{
cout << "You have cold";
}
  else if(fever== "COMMON" && fatigue=="COMMON" && cough== "COMMON" && sneezing == "NO" && aches== "COMMON" && nose=="SOMETIMES" && throat=="SOMETIMES" && diarea=="SOMETIMES" && headache=="COMMON" && breath=="NO")
{
cout << "You have Flu";
}
else{
	cout << "You dont have any of 3 viruses";
}


}
```

#4 MAX SUM

```
#include<bits/stdc++.h>
```

```
#include<math.h>
#include<cmath>
using namespace std;
int main()
{
   long int n,k,temp,sum=0;
   cin>>n;
   cin>>k;
   vector<int> v;
   for(int i=0;i<n;i++)
   {
       cin>>temp;
       sum=sum + temp;
       v.push_back(temp);
   }
   make_heap(v.begin(),v.end());
   long int maxi = 0,res = 0;
   for(int i=0;i<k;i++)
   {
      maxi=v.front();
      sum-=maxi;
      pop_heap(v.begin(), v.end());
      v.pop_back();
      res = maxi / 2;
      sum+=res;
      v.push_back(res);
   push_heap(v.begin(),v.end());
   }
   cout<<sum;
}
```

```
#5 Single lane highway
from itertools import permutations
import math

# def get_count(d):
#     c=0
#     for i in d:
#         c+=1
#     return c

n=int(input())
l=list(map(int,input().split()))

cc=[]

# d1=permutations(l,n-1)
# d2=permutations(l,n)
# cc.append(get_count(d1))
# cc.append(get_count(d2))

s1=math.factorial(n)//math.factorial(n-(n))
s2=math.factorial(n)//math.factorial(n-(n-1))

cc.append(s1)
cc.append(s2)

if(n%2==0):
   t=sum(cc)+2
else:
   t=sum(cc)-1
```

```
print("%.6f"%(t/cc[-1]))
```

#6TANIC

```
#include <bits/stdc++.h>
using namespace std;
#define ull      unsigned long long int
#define ll       long long int
#define loop(i,s,e)     for(ll i=(s);i<(e);i++)
#define rloop(i,s,e)    for(ll i=(s);i>=(e);i--)
#define scan(any)       for(auto &i:any) cin>>i;
#define print(any)      for(auto i:any) cout<<i<<" "; nl;
#define nl              cout<<'\n'
 #define pi 3.141592654
#define hell 1000000007
#define io ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0)
#define fix(n) cout << fixed << setprecision(n)
#define input1(n) int n;cin>>n
#define input2(a, b) int a,b;cin>>a>>b
#define Max(a,b) ((a)>(b)?(a):(b))
#define Min(a,b) ((a)<(b)?(a):(b))
#define rep(i,a,b) for (__typeof((b)) i=(a);i<(b);i++)
#define ren(i,a,b) for(__typeof((a)) i=(a);i>=(b);i--)
#define mp make_pair
#define pb push_back
#define fi first
#define se second
#define vi vector<int>
#define pii pair<int,int>
#define piii pair<pair<int,int>,int>
#define all(v) (v).begin(), (v).end()
#define sz(x) (int)x.size()
```

```
#define set(a,n) memset(a,n,sizeof(a))
void calc(int i,vi &v1,int siz,int s,int &tot)
{
if(i==siz)
{
if(s==0)
tot++;
return;
}

calc(i+1,v1,siz,s+v1[i],tot);
calc(i+1,v1,siz,s,tot);
}



int main()
{
int n;
cin>>n;
vi v(n);
scan(v);
int m=0;

for(int i=0;i<n;i++)
{
if(v[i]>m)
m=v[i];
}
int count=0;
while(m)
{
```

```
count++;
m=m>>1;
}
vi v1(n,0);
for(int i=0;i<n;i++)
{
while(v[i])
{
if(v[i]&1)
v1[i]++;
v[i]=v[i]>>1;
}
}
int j=0;
for(int i=0;i<n;i++)
{
v1[j]=count-2*v1[i];
if(v1[j]==0)
continue;
else
j++;
}
int tot=0;
calc(0,v1,j,0,tot);
tot-=1;
tot=tot*(1+n-j)+(1<<(n-j))-1;
    vi bin(count,0);
    int i=0;
    while (tot > 0) {

        bin[i] = tot &1;
        tot = tot>>1;
    i++;
    }
    for (int j = count - 1; j >= 0; j--)
        cout << bin[j];
return 0;
}
```

#7 FACTOR THREE(3)

```
for _ in range(int(input())):
    n=int(input())
    l=list(map(int,input().split()))
    a=[]
    for i in range(n):
        a.append(l[i]%3)
    z=a.count(0)
    o=a.count(1)
    t=a.count(2)
    if z==0 and o!=0 and t!=0:
        print('NO')
    elif z==0 and t==0 and o!=0:
        print('YES')
    elif z==0 and o==0 and t!=0:
        print('YES')
    elif z<=(t+o):
        print('YES')
    else:
        print('NO')
```

#8 FILL CUBE

```
import math
n=int(input())
l=[]
```

```
c=0
for i in range(n):
   l.append(list(map(str,input().split())))
for j in range(n):
   for k in range(n):
      if l[j][k]=='D':
         c+=1

print(math.floor(math.sqrt(c)))
```

```
#9 EVEN ODD
from itertools import product
def sum_of_tup(n):
   sum=0
   for i in range(len(n)):
      sum=sum+int(n[i])
   return sum
low,high=map(int,input().split())
k=int(input())
lst=[]
for i in range(low,high+1):
   lst.append(str(i))
count=0
perm=product(lst,repeat=k)
for i in perm:
   if (sum_of_tup(i)%2==0):
      count+=1
print(count%1000000007)
```

```
#10 PARTICALE


#include<bits/stdc++.h>
```

```
#include<math.h>

#include<cmath>

using namespace std;

float dist(float x1,float x2,float y1,float y2,float z1,float z2){

  float d=0;

  d=sqrt(pow(x2 - x1, 2) +  pow(y2 - y1, 2) +  pow(z2 - z1, 2) * 1.0);

  return d;

}

float area(float side1, float side2, float side3 ){

  float s = (side1+side2+side3)/2;

  float are = sqrt(s*(s-side1)*(s-side2)*(s-side3));

    return are;

}

 int main(){

   float h,a,b,c,d,va,vb,vc,vd;

   cin>>h>>a>>b>>c>>d>>va>>vb>>vc>>vd;

  char da,db,dc,dd;

  cin>>da>>db>>dc>>dd;

  if(da=='D'){

    va=va*(-1);

  }

  if(db=='D'){

    vb=vb*(-1);

  }

  if(dc=='D'){

    vc=vc*(-1);

  }

  if(dd=='D'){

    vd=vd*(-1);

  }

  float xa=0,ya=h*(-1);
  float xb=h,yb=h*(-1);
  float xc=h,yc=0;
  float xd=0,yd=0;
  float z[100][4];
  for( int i=0;i<100;i++){
    for( int j=0;j<4;j++){
      z[i][j]=0;
    }
  }
  z[0][0]=a;
  z[0][1]=b;
  z[0][2]=c;
  z[0][3]=d;
  for( int i=1;i<100;i++){
    z[i][0]=z[i-1][0]+va;
    z[i][1]=z[i-1][1]+vb;
    z[i][2]=z[i-1][2]+vc;
    z[i][3]=z[i-1][3]+vd;
    if(z[i][0] > h){
      z[i][0]=h;
    }
    if(z[i][0] < 0){
      z[i][0]=0;
    }
    if(z[i][1] > h){
      z[i][1]=h;
    }
    if(z[i][1] < 0){
      z[i][1]=0;
    }
    if(z[i][2] > h){
      z[i][2]=h;
    }
    if(z[i][2] < 0){
      z[i][2]=0;
    }
    if(z[i][3] > h){
      z[i][3]=h;
    }
    if(z[i][3] < 0){
      z[i][3]=0;
    }
  }
  float ab[100];
  for( int i=0;i<100;i++){
    ab[i]=dist(xa,xb,ya,yb,z[i][0],z[i][1]);
  }
  float bc[100];
  for( int i=0;i<100;i++){
    bc[i]=dist(xb,xc,yb,yc,z[i][1],z[i][2]);
  }
   float ac[100];
  for( int i=0;i<100;i++){
    ac[i]=dist(xa,xc,ya,yc,z[i][0],z[i][2]);
  }
   float ad[100];
  for( int i=0;i<100;i++){
    ad[i]=dist(xa,xd,ya,yd,z[i][0],z[i][3]);
  }
  float bd[100];
  for( int i=0;i<100;i++){
    bd[i]=dist(xb,xd,yb,yd,z[i][1],z[i][3]);
```

```
}
 float cd[100];
for( int i=0;i<100;i++){
  cd[i]=dist(xc,xd,yc,yd,z[i][2],z[i][3]);
}


float abc[100];
for(int i=0;i<100;i++){
  abc[i]=area(ab[i],bc[i],ac[i]);
}
float adc[100];
for(int i=0;i<100;i++){
  adc[i]=area(ad[i],cd[i],ac[i]);
}
float abd[100];
for(int i=0;i<100;i++){
  abd[i]=area(ab[i],ad[i],bd[i]);
}
float bcd[100];
for(int i=0;i<100;i++){
  bcd[i]=area(bc[i],cd[i],bd[i]);
}


float maxabc = abc[0];
  for (int i = 0; i < 100; i++){
          if (maxabc < abc[i])
        maxabc = abc[i];
}
  float minabc = abc[0];
      for (int i = 0; i < 100; i++)
      {
```

```
      if (minabc > abc[i])
        minabc = abc[i];
    }
  float maxadc = adc[0];
    for (int i = 0; i < 100; i++)
    {
      if (maxadc < adc[i])
       maxadc = adc[i];
    }
  float minadc = adc[0];
    for (int i = 0; i < 100; i++)
    {
      if (minadc > adc[i])
        minadc = adc[i];
   }
  float ans1=4*pow((maxabc+maxadc),2);
  float ans2=4*pow((minabc+minadc),2);
    cout<<round(ans1)<<" "<<round(ans2)<<endl;
 return 0;

}
```
