

## 最小循环子数组

### 题目描述：

给定一个由若干整数组成的数组  $nums$ ，请检查数组是否是由某个子数组重复循环拼接而成，请输出这个最小的子数组。

### 输入描述：

第一行输入数组中元素个数  $n$ ， $1 \leq n \leq 100000$

第二行输入数组的数字序列  $nums$ ，以空格分割， $0 \leq nums[i] < 10$ ;

### 输出描述：

输出最小的子数组的数字序列，以空格分割；

### 补充说明：

数组本身是其最大的子数组，循环  $1$  次可生成的自身；

### 示例 1

#### 输入：

```
9
1 2 1 1 2 1 1 2 1
```

#### 输出：

```
1 2 1
```

#### 说明：

数组  $[1,2,1,1,2,1,1,2,1]$  可由子数组  $[1,2,1]$  重复循环  $3$  次拼接而成

```
#include<bits/stdc++.h>
using namespace std;
int equaln(string temp,string str){
    int c=0;
    string s;
    int n=str.length()/temp.length();
    for(int i=0;i<n;i++){
        s+=temp;
    }
    for(int i=0;i<s.length();i++){
        if(s[i]==str[i])c++;
    }
    if(c==s.length())return 1;
    else
        return 0;
```

```

}
int equal(string s,string str){
    int c=0;
    for(int i=0;i<s.length();i++){
        if(s[i]==str[i])c++;
    }
    if(c==s.length())return 1;
    else
        return 0;

}
int check(string temp,string str){
    int n=0;
    int n=str.length()/temp.length();
    string s="";
    for(int i=0;i<n;i++){
        s+=temp;
        if(equal(s,str))return 1;
    }
    return 0;
}
int main(){
    int n;cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    string str="";
    for(int i=0;i<n;i++){
        str+=char(a[i]+'0');
    }

    int len=str.length();
    string temp;
    int min=100000;
    string result="";
    int r=100000;

    for(int i=0;i<str.length();i++){
        if(str.length()%(i+1)==0)

```

```

        //for(int j=i;j<str.length();j++)
    {
        if(str.length()%(i+1)==0){
            temp=str.substr(0,i+1);
            if(equalN(temp,str)){
                result=temp;
                for(int i=0;i<result.length();i++){
                    cout<<result[i]<<" ";
                }
                return 0;
            }
        }
    }
    for(int i=0;i<result.length();i++)
        cout<<result[i]<<" ";
    return 0;
}

```