# TIMUS 1595 Perfect Sequence Solution

write by Gestalti Lur

2012-09-25

題目鏈接：TIMUS 1595

## 題目大意

要求構造一個滿足如下條件的序列：

1.包含數字 1..N(N<=250000)
2.長度不超過 2*N
3.所有的數字都是不同的且小於 10^13
4.前 K 個數字的和能夠被 K 整除

## 算法分析

如果設前 i 個數字的和爲 S_i,那麼必然有 S_i mod i = 0.由於 S_i 單調增，所以問題也就等同於找出一個沒有出現過的 q=(S_i-S_i-1)。這樣必然會在 2*N-1 的長度內構造出該序列[1]。設 S_i=min(i*p(i*p>S_i-1))。如果 S_i 小於 N,那麼可以直接 hash,否則可以記錄一個之前選取的大於 N 的數字中最小的數，對於每次大於 N 的情況，選取這個大於這個數字的 i*p 就不會有重複，N 的範圍相比於 10^13 還是微不足道的。

## 參考代碼

```
/*
  TMUS 1595
  gestapolur
  ACCEPTED
  2012-09-25
*/
#include<cstdio>
#include<cstring>
#define MAXN 250013

long long a[ 2 * MAXN ];
int n , m ;
long long sum;
bool h[ MAXN ];

int min( int a , int b ){ return a < b ? a : b ;}

void constr()
{
  int cnt = 2 , k = 2;
  long long t , i , bigmin = n + 1;
```

[1] 測試結果如此，沒有經過證明。

```
 //memset( h , false , sizeof( h ) );
 h[ 1 ] = true;
 sum = 1;
 a[ 1 ] = 1;
 while( cnt <= n )
   {
    i = sum / k + 1;
    while( i * k - sum <= n and h[ i * k - sum ] ) ++ i;
    t = i * k - sum;
    if( t > n )
     {
      while( t < bigmin ) t += k;
      bigmin = t;
     }
    sum = i * k;
    a[ k ++ ] = t;
    if( t <= n ) { ++ cnt; h[ t ] = true;}
   }
 m = k - 1;
 return ;
}

int main()
{
 scanf( "%d" , &n );
 constr();
 for( int i = 1 ; i <= m ; ++ i ) printf("%lld " , a[ i ] ); printf("\n");
 return 0;
}
```