Gmail Neil Sloane <njasloane@gmail.com>

Re: OeisWiki email from user "N. J. A. Sloane"

Zhaohui Du <zhao.hui.du@gmail.com> Sun, Jul 3, 2022 at 7:36 PM
To: "N. J. A. Sloane" <njasloane@gmail.com>

One of the broken link is for the output result and it should just provide the sequence so we could remove it.
Another one is for the C++ source code. There're copy of the code in https://bbs.emath.ac.cn/thread-705-16-1.html
Two C++ windows code is used and according to the introduction, it ran for 10hours and generated 15.6G data in disk to process N from 1 to 11
and it ran for 37hours and generated 54G data in disk. But no introduction for N=13 is available.

The first piece of code is:

```
#include "stdafx.h"
#define N 8
#include <set>
using namespace std;

class number{
    int up;
    int down;
public:
    number(const number& n):up(n.up),down(n.down){}
    number(int n=0):up(n),down(1){}
    number& operator+=(const number& n);
    number& operator-=(const number& n);
    number& operator*=(const number& n);
    number& operator/=(const number& n);
    bool is_zero()const{return down!=0&&up==0;}
    bool is_valid()const{return down!=0;}
    bool is_one()const{return down!=0&&up==down;}
    bool operator==(const number& n)const{return
        is_valid()&&n.is_valid()&&n.down*(long long)up==n.up*(long long)down;}
    bool operator<(const number& n)const;
    bool operator>(const number& n)const{return n<*this;}
    bool operator<=(const number& n)const{return !(*this>n);}
    bool operator!=(const number& n)const{return !(n==*this);}
    bool operator>=(const number& n)const{return !(*this<n);}
    number operator+(const number& n)const{number m(*this);return m+=n;}
    number operator-(const number& n)const{number m(*this);return m-=n;}
    number operator*(const number& n)const{number m(*this);return m*=n;}
    number operator/(const number& n)const{number m(*this);return m/=n;}
    bool is_integer()const{return down!=0&&up%down==0;}
    int get_integer()const{if(is_integer())return up/down;else return -1;}
    number& operator=(int n){up=n;down=1;return *this;}
    int get_up()const{return up;}
    int get_down()const{return down;}
        number abs()const{number x;x.up=up>=0?up:-up;x.down=down>=0?down:-down;return x;}
};

number& number::operator +=(const number& n)
{
    up=up*n.down+down*n.up;
    down*=n.down;
    return *this;
}

number& number::operator -=(const number& n)
```

```
{
    up=up*n.down-down*n.up;
    if(up<0)up=-up;
    down*=n.down;
    return *this;
}

number& number::operator *=(const number& n)
{
    up*=n.up;
    down*=n.down;
    return *this;
}

number& number::operator /=(const number& n)
{
    down*=n.up;
    up*=n.down;
    return *this;
}

bool number::operator <(const number &n)const
{
    return (long long)up*n.down<n.up*(long long)down;
}

bool has_set(int x[],int size)
{
    int mask=0;
    int i;
    char fName[20];
    for(i=0;i<size;i++){
        mask|=1<<(x[i]-1);
    }
    sprintf(fName,"data\\%d",mask);
    FILE *pFile=fopen(fName,"rb");
    if(pFile==NULL)
        return false;
    fclose(pFile);
    return true;
}

#define BUFF_LEN 4096
number buff[BUFF_LEN];
number buff2[BUFF_LEN];

void output_r(int i)
{
    int s=(1<<i)-1;
    char fName[20];
    sprintf(fName,"data\\%d",s);
    FILE *pFile=fopen(fName,"rb");
    if(pFile==NULL){
        fprintf(stderr,"Cannot read file %s\n",fName);
        exit(-1);
    }
    int yu=fread(buff,sizeof(number),BUFF_LEN,pFile);
    int x=1,start;
    if(buff[0]==0)start=1;else start=0;
    while(buff[start]<=x){
```

```
                if(buff[start]<x){
                        start++;
                }else{
                        x++;
                        start++;
                }
        if(start>=yu){
            if(yu==BUFF_LEN){
                yu=fread(buff,sizeof(number),BUFF_LEN,pFile);
                start=0;
                if(yu==0)break;
            }else break;
        }
    }
    printf("r[%d]=%d\n",i,x);
    fclose(pFile);
}

void save_set(const set<number>& result, int x[],int size)
{
    int mask=0;
    int i;
    char fName[20];
    for(i=0;i<size;i++){
        mask|=1<<(x[i]-1);
    }
    sprintf(fName,"data\\%d",mask);
    FILE *pFile=fopen(fName,"wb");
    if(pFile==NULL){
        fprintf(stderr,"Cannot write file %s\n",fName);
        exit(-1);
    }
    int j;
    set<number>::const_iterator cit;
    i=j=0;
    for(cit=result.begin();cit!=result.end();++cit){
        buff[j++]=*cit;
        i++;
        if(j==BUFF_LEN){
            j=0;
            fwrite(buff,sizeof(number),BUFF_LEN,pFile);
        }
    }
    if(j>0){
        fwrite(buff,sizeof(number),j,pFile);
    }
    fclose(pFile);

    if((mask&(mask+1))==0){
        output_r(size);
    }
}


void add_result(set<number>& result, int y[],int yc,int z[],int zc)
{
    int mask=0;
    int i,j;
    char fName[20];
    FILE *yFile,*zFile;
```

```
166.      int yu,zu;
167.      if(yc<=2){
168.          if(yc==1){
169.              yu=1;
170.              buff[0]=y[0];
171.          }else{
172.              yu=5;
173.              buff[0]=y[0]+y[1];
174.              buff[1]=abs(y[0]-y[1]);
175.              buff[2]=y[0]*y[1];
176.                          buff[3]=number(y[0])/y[1];
177.                          buff[4]=number(y[1])/y[0];
178.          }
179.          yFile=NULL;
180.      }else{
181.          mask=0;
182.          for(i=0;i<yc;i++){
183.              mask|=1<<(y[i]-1);
184.          }
185.          sprintf(fName,"data\\%d",mask);
186.          yFile=fopen(fName,"rb");
187.          if(yFile==NULL){
188.              fprintf(stderr,"Cannot read file %s\n",fName);
189.              exit(-1);
190.          }
191.          yu=fread(buff,sizeof(number),BUFF_LEN,yFile);
192.      }
193.      if(zc<=2){
194.          if(zc==1){
195.              zu=1;
196.              buff2[0]=z[0];
197.          }else{
198.              zu=5;
199.              buff2[0]=z[0]+z[1];
200.              buff2[1]=abs(z[0]-z[1]);
201.              buff2[2]=z[0]*z[1];
202.                          buff2[3]=number(z[0])/z[1];
203.                          buff2[4]=number(z[1])/z[0];
204.          }
205.          zFile=NULL;
206.      }else{
207.          mask=0;
208.          for(i=0;i<zc;i++){
209.              mask|=1<<(z[i]-1);
210.          }
211.          sprintf(fName,"data\\%d",mask);
212.          zFile=fopen(fName,"rb");
213.          if(zFile==NULL){
214.              fprintf(stderr,"Cannot read file %s\n",fName);
215.              exit(-1);
216.          }
217.          zu=fread(buff2,sizeof(number),BUFF_LEN,zFile);
218.      }
219.      if(zu<BUFF_LEN){
220.          do{
221.              for(i=0;i<zu;i++)for(j=0;j<yu;j++){
222.                  result.insert(buff2[i]+buff[j]);
223.                  result.insert(buff2[i]*buff[j]);
224.                  result.insert((buff2[i]-buff[j]).abs());
225.                                  number x=buff2[i]/buff[j];
```

```
                                if(x.is_valid()){
                                        result.insert(x);
                                }
                                x=buff[j]/buff2[i];
                                if(x.is_valid()){
                                        result.insert(x);
                                }
            }
            if(yu<BUFF_LEN)break;
            yu=fread(buff,sizeof(number),BUFF_LEN,yFile);
        }while(1);
    }else if(yu<BUFF_LEN){
        do{
            for(i=0;i<zu;i++)for(j=0;j<yu;j++){
                result.insert(buff2[i]+buff[j]);
                result.insert(buff2[i]*buff[j]);
                result.insert((buff2[i]-buff[j]).abs());
                                number x=buff2[i]/buff[j];
                                if(x.is_valid()){
                                        result.insert(x);
                                }
                                x=buff[j]/buff2[i];
                                if(x.is_valid()){
                                        result.insert(x);
                                }
            }
            if(zu<BUFF_LEN)break;
            zu=fread(buff2,sizeof(number),BUFF_LEN,zFile);
        }while(1);
    }else{
        do{
            do{
                for(i=0;i<zu;i++)for(j=0;j<yu;j++){
                    result.insert(buff2[i]+buff[j]);
                    result.insert(buff2[i]*buff[j]);
                    result.insert((buff2[i]-buff[j]).abs());
                                        number x=buff2[i]/buff[j];
                                        if(x.is_valid()){
                                                result.insert(x);
                                        }
                                        x=buff[j]/buff2[i];
                                        if(x.is_valid()){
                                                result.insert(x);
                                        }
                }
                if(yu<BUFF_LEN)break;
                yu=fread(buff,sizeof(number),BUFF_LEN,yFile);
            }while(1);
            if(zu<BUFF_LEN)break;
            zu=fread(buff2,sizeof(number),BUFF_LEN,zFile);
            if(zu==0)break;
            fseek(yFile,0,SEEK_SET);
            yu=fread(buff,sizeof(number),BUFF_LEN,yFile);
        }while(1);
    }
    if(yFile)fclose(yFile);
    if(zFile)fclose(zFile);
}
```

```
void gen_set(int x[],int size)
{
    int y[N],z[N];
    int i,j,yc,zc;
    if(size<=2)return;
    if(has_set(x,size))return;
    set<number> result;
    for(i=1;i<(1<<size)-1;i+=2){
        yc=zc=0;
        for(j=0;j<size;j++){
            if(i&(1<<j)){
                y[yc++]=x[j];
            }else{
                z[zc++]=x[j];
            }
        }
        gen_set(y,yc);
        gen_set(z,zc);
        add_result(result,y,yc,z,zc);
    }
    save_set(result,x,size);
}

int bc(int x){
        int c=0;
        while(x>0){
                if(x&1)c++;
                x>>=1;
        }
        return c;
}

int _tmain(int argc, _TCHAR* argv[])
{
    system("mkdir data");
    int x[]={1,2,3,4,5,6,7,8};
        int i;
        for(i=7;i<1<<12;i++){
                if(bc(i)<=8){
                        int j=0,k;
                        for(k=0;k<12;k++){
                                if(i&(1<<k)){
                                        x[j++]=k+1;
                                }
                        }
                        gen_set(x,j);
                }
        }
        return 0;
}
```

The 2nd piece of code is:

```
// checker.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include <algorithm>
using namespace std;
```

```
#include <stdio.h>
#include <stdlib.h>

class number{
    int up;
    int down;
public:
    number(int u,int d):up(u),down(d){}
    number(const number& n):up(n.up),down(n.down){}
    number(int n=0):up(n),down(1){}
    number& operator+=(const number& n);
    number& operator-=(const number& n);
    number& operator*=(const number& n);
    number& operator/=(const number& n);
    bool is_zero()const{return down!=0&&up==0;}
    bool is_valid()const{return down!=0;}
    bool is_one()const{return down!=0&&up==down;}
    bool operator==(const number& n)const{return
        is_valid()&&n.is_valid()&&n.down*(long long)up==n.up*(long long)down;}
    bool operator<(const number& n)const;
    bool operator>(const number& n)const{return n<*this;}
    bool operator<=(const number& n)const{return !(*this>n);}
    bool operator!=(const number& n)const{return !(n==*this);}
    bool operator>=(const number& n)const{return !(*this<n);}
    number operator+(const number& n)const{number m(*this);return m+=n;}
    number operator-(const number& n)const{number m(*this);return m-=n;}
    number operator*(const number& n)const{number m(*this);return m*=n;}
    number operator/(const number& n)const{number m(*this);return m/=n;}
    bool is_integer()const{return down!=0&&up%down==0;}
    int get_integer()const{if(is_integer())return up/down;else return -1;}
    number& operator=(int n){up=n;down=1;return *this;}
    int get_up()const{return up;}
    int get_down()const{return down;}
};

number& number::operator +=(const number& n)
{
    up=up*n.down+down*n.up;
    down*=n.down;
    return *this;
}

number& number::operator -=(const number& n)
{
    up=up*n.down-down*n.up;
    if(up<0)up=-up;
    down*=n.down;
    return *this;
}

number& number::operator *=(const number& n)
{
    up*=n.up;
    down*=n.down;
    return *this;
}

number& number::operator /=(const number& n)
{
    down*=n.up;
```

```
    up*=n.down;
    return *this;
}

bool number::operator <(const number &n)const
{
    return (long long)up*n.down<n.up*(long long)down;
}

long long gcd(long long x,long long y)
{
    if(x==0)return y;
    return gcd(y%x,x);
}

class lnumber{
    long long up;
    long long down;
public:
    lnumber(const lnumber& n):up(n.up),down(n.down){}
    lnumber(const number& n):up(n.get_up()),down(n.get_down()){}
    lnumber(long long n=0):up(n),down(1LL){}
    lnumber& operator+=(const lnumber& n);
    lnumber& operator-=(const lnumber& n);
    lnumber& operator*=(const lnumber& n);
    lnumber& operator/=(const lnumber& n);
    bool is_zero()const{return down!=0&&up==0;}
    bool is_valid()const{return down!=0;}
    bool is_one()const{return down!=0&&up==down;}
    void normalize(){
        long long g=gcd(up,down);
        up/=g;down/=g;
    }
    bool operator==(const lnumber& n)const{
        if(!is_valid()||!n.is_valid())
            return false;
        if(n.down*up!=n.up*down)
            return false;
        ///check for overflow
        long long d1=n.down%0x7FFFFFFF;
        long long u1=n.up%0x7FFFFFFF;
        long long d2=down%0x7FFFFFFF;
        long long u2=up%0x7FFFFFFF;
        if((d1*u2)%0x7FFFFFFF!=(d2*u1)%0x7FFFFFFF)
            return false;
        return true;
    }
    bool operator<(const lnumber& n)const;
    bool operator>(const lnumber& n)const{return n<*this;}
    bool operator<=(const lnumber& n)const{return !(*this>n);}
    bool operator!=(const lnumber& n)const{return !(n==*this);}
    bool operator>=(const lnumber& n)const{return !(*this<n);}
    lnumber operator+(const lnumber& n)const{lnumber m(*this);return m+=n;}
    lnumber operator-(const lnumber& n)const{lnumber m(*this);return m-=n;}
    lnumber operator*(const lnumber& n)const{lnumber m(*this);return m*=n;}
    lnumber operator/(const lnumber& n)const{lnumber m(*this);return m/=n;}
    bool is_integer()const{return down!=0&&up%down==0;}
    long long get_integer()const{if(is_integer())return up/down;else return -1LL;}
    lnumber& operator=(long long n){up=n;down=1LL;return *this;}
    long long get_up()const{return up;}
```

```
    long long get_down()const{return down;}
};

lnumber& lnumber::operator +=(const lnumber& n)
{
    up=up*n.down+down*n.up;
    down*=n.down;
    return *this;
}

lnumber& lnumber::operator -=(const lnumber& n)
{
    up=up*n.down-down*n.up;
    if(up<0)up=-up;
    down*=n.down;
    return *this;
}

lnumber& lnumber::operator *=(const lnumber& n)
{
    up*=n.up;
    down*=n.down;
    return *this;
}

lnumber& lnumber::operator /=(const lnumber& n)
{
    down*=n.up;
    up*=n.down;
    return *this;
}

#define M32 0xFFFFFFFF
void mult128(unsigned int out[4],long long in1, long long in2)
{
    unsigned high1,low1,high2,low2;
    high1=(in1>>32)&M32;
    low1=in1&M32;
    high2=(in2>>32)&M32;
    low2=in2&M32;
    unsigned long long r1,r2,r3,r4;
    r1=low1*low2;
    r2=low1*high2;
    r3=high1*low2;
    r4=high1*high2;
    out[0]=(unsigned)(r1&M32);
    unsigned long long u=(r1>>32)&M32;
    u+=(r2&M32)+(r3&M32);
    out[1]=(unsigned)(u&M32);
    u=u>>32;
    u+=(r2>>32)+(r3>>32);
    u+=r4&M32;
    out[2]=(unsigned)(u&M32);
    u=u>>32;
    u+=(r4>>32);
    out[3]=(unsigned)u;
}

bool lnumber::operator <(const lnumber &n)const
{
```

```
    unsigned int r1[4],r2[4];
    int i;
    mult128(r1,up,n.down);
    mult128(r2,n.up,down);
    for(i=3;i>=0;i--){
        if(r1[i]<r2[i])
            return true;
        if(r1[i]>r2[i])
            return false;
    }
}

#define DATA_PATH "..\\..\\unr2\\unr2\\data\"
#define FILT_PATH "..\\..\\unr2\\unr2\\idata\"
#define T 38103

int bc(int x){
        int c=0;
        while(x>0){
                if(x&1)c++;
                x>>=1;
        }
        return c;
}

#define BUFF_LEN 4096
#define MAX_BUFF_SIZE (40*1024*1024)
number buff[BUFF_LEN];
number buff2[MAX_BUFF_SIZE];

bool test(number buff2[],int zu, const lnumber& r)
{
    lnumber rr=r;rr.normalize();
    if(rr.get_up()>0x7FFFFFFF||rr.get_down()>0x7FFFFFFF)
        return false;
    number sr((int)rr.get_up(),(int)rr.get_down());
    return binary_search(buff2,buff2+zu,sr);
}


///x is small set, y is large set
bool verify(int x[],int xc, int y[],int yc, const lnumber& result)
{
    int mask1,mask2;
    char f1Name[100],f2Name[100];
    int i;
    mask1=mask2=0;
    for(i=0;i<xc;i++){
        mask1|=1<<(x[i]-1);
    }
    for(i=0;i<yc;i++){
        mask2|=1<<(y[i]-1);
    }
    sprintf(f1Name,DATA_PATH "%d",mask1);
    sprintf(f2Name,DATA_PATH "%d",mask2);
    FILE *f1,*f2;
    f1=f2=NULL;
    int yu,zu;
    if(xc>2){
        f1=fopen(f1Name,"rb");
```

```
247.          if(f1==NULL){
248.              fprintf(stderr,"Cannot open file %s\n",f1Name);
249.              exit(-1);
250.          }
251.          yu=fread(buff,sizeof(number),BUFF_LEN,f1);
252.      }else{
253.          if(xc==1){
254.              yu=1;
255.              buff[0]=x[0];
256.          }else{
257.              yu=5;
258.              buff[0]=x[0]+x[1];
259.              buff[1]=abs(x[0]-x[1]);
260.              buff[2]=x[0]*x[1];
261.              buff[3]=number(x[0])/x[1];
262.              buff[4]=number(x[1])/x[0];
263.          }
264.      }
265.      f2=fopen(f2Name,"rb");
266.      if(f2==NULL){
267.          fprintf(stderr,"Cannot open file %s\n",f2Name);
268.          exit(-1);
269.      }
270.      zu=fread(buff2,sizeof(number),MAX_BUFF_SIZE,f2);
271.      fclose(f2);
272.
273.      do{
274.          for(i=0;i<yu;i++){
275.              lnumber lx=buff[i];
276.              lnumber s=lx+result;
277.              if(test(buff2,zu,s)){
278.                  if(f1)fclose(f1);
279.                  return true;
280.              }
281.              s=lx-result;
282.              if(test(buff2,zu,s)){
283.                  if(f1)fclose(f1);
284.                  return true;
285.              }
286.              if(!lx.is_zero()){
287.                  s=lx*result;
288.                  if(test(buff2,zu,s)){
289.                      if(f1)fclose(f1);
290.                      return true;
291.                  }
292.              }
293.              if(result.is_zero()&&lx.is_zero()){
294.                  if(f1)fclose(f1);
295.                  return true;
296.              }
297.              if(!lx.is_zero()){
298.                  s=result/lx;
299.                  if(test(buff2,zu,s)){
300.                      if(f1)fclose(f1);
301.                      return true;
302.                  }
303.              }
304.              if(!lx.is_zero()&&!result.is_zero()){
305.                  s=lx/result;
306.                  if(test(buff2,zu,s)){
```

```
                    if(f1)fclose(f1);
                    return true;
                }
            }
        }
        if(yu<BUFF_LEN)break;
        yu=fread(buff,sizeof(number),BUFF_LEN,f1);
    }while(1);

    if(f1)fclose(f1);
    return false;
}

bool vs(int x[],int c, lnumber result)
{
        result.normalize();
        if(result.get_up()>=0x7FFFFFFF||result.get_down()>=0x7FFFFFFF)
                return false;
        number sr((int)result.get_up(),(int)result.get_down());
    int mask;
    char fName[100];
    int i;
    mask=0;
    for(i=0;i<c;i++){
        mask|=1<<(x[i]-1);
    }
    sprintf(fName,DATA_PATH "%d",mask);
    FILE *f;
    f=NULL;
    int yu;
    if(c>2){
        f=fopen(fName,"rb");
        if(f==NULL){
            fprintf(stderr,"Cannot open file %s\n",fName);
            exit(-1);
        }
        yu=fread(buff,sizeof(number),BUFF_LEN,f);
    }else{
        if(c==1){
            yu=1;
            buff[0]=x[0];
        }else{
            yu=5;
            buff[0]=x[0]+x[1];
            buff[1]=abs(x[0]-x[1]);
            buff[2]=x[0]*x[1];
            buff[3]=number(x[0])/x[1];
            buff[4]=number(x[1])/x[0];
        }
    }

        do{
                for(i=0;i<yu;i++){
                        if(buff[i]==sr){
                                if(f)fclose(f);
                                return true;
                        }
                }
        if(yu<BUFF_LEN)break;
        yu=fread(buff,sizeof(number),BUFF_LEN,f);
```

```
    }while(1);
        if(f)fclose(f);
        return false;
}

bool v9(int x[],lnumber result)
{///verify 9 elements in x could get 'result'
    int u;
    int i,j;
    int N=9;
    int uc,vc;
    int au[9],av[9];
    for(u=(N+1)/2;u<=8;u++){
        int v=N-u;
        for(i=1;i<(1<<N)-1;i++){
            if(bc(i)==v){
                uc=vc=0;
                for(j=0;j<N;j++){
                    if(i&(1<<j)){
                        av[vc++]=j+1;
                    }else{
                        au[uc++]=j+1;
                    }
                }
                if(verify(av,vc,au,uc,result))
                    return true;
            }
        }
    }
    return false;
}

bool v10(int x[],lnumber result)
{
    int u;
    int i,j;
    int N=10;
    int uc,vc;
    int au[10],av[10];
    for(u=(N+1)/2;u<=8;u++){
        int v=N-u;
        for(i=1;i<(1<<N)-1;i++){
            if(bc(i)==v){
                uc=vc=0;
                for(j=0;j<N;j++){
                    if(i&(1<<j)){
                        av[vc++]=x[j];
                    }else{
                        au[uc++]=x[j];
                    }
                }
                if(verify(av,vc,au,uc,result))
                    return true;
            }
        }
    }
    for(u=0;u<10;u++){
        int h=x[u];
        uc=0;
        for(i=0;i<10;i++){
```

```
            if(i!=u){
                au[uc++]=x[i];
            }
        }
        if(v9(au,result+h))
            return true;
        if(v9(au,result-h))
            return true;
        if(result.is_zero()&&h==0)return true;
        if(h!=0){
            if(v9(au,result*h))
                return true;
            if(v9(au,result/h))
                return true;
            if(!result.is_zero()){
                if(v9(au,lnumber(h)/result))
                    return true;
            }
        }
    }
    return false;
}

bool v11(int x[],lnumber result)
{
    int u;
    int i,j;
    int N=11;
    int uc,vc;
    int au[11],av[11];
    for(u=(N+1)/2;u<=8;u++){
        int v=N-u;
        for(i=1;i<(1<<N)-1;i++){
            if(bc(i)==v){
                uc=vc=0;
                for(j=0;j<N;j++){
                    if(i&(1<<j)){
                        av[vc++]=x[j];
                    }else{
                        au[uc++]=x[j];
                    }
                }
                if(verify(av,vc,au,uc,result))
                    return true;
            }
        }
    }
    for(u=0;u<11;u++){
        int h=x[u];
        uc=0;
        for(i=0;i<11;i++){
            if(i!=u){
                au[uc++]=x[i];
            }
        }
        if(v10(au,result+h))
            return true;
        if(v10(au,result-h))
            return true;
        if(result.is_zero()&&h==0)return true;
```

```
        if(h!=0){
            if(v10(au,result*h))
                return true;
            if(v10(au,result/h))
                return true;
            if(!result.is_zero()){
                if(v10(au,lnumber(h)/result))
                    return true;
            }
        }
                int u2;
                for(u2=u+1;u2<11;u2++){
                        int hc=5,h2=x[u2];
                        number hh[5];
                        hh[0]=h+h2;hh[1]=abs(h-h2);hh[2]=h*h2;hh[3]=number(h,h2);hh[4]=number(h2,h);
                        for(int k=0;k<5;k++){
                                uc=0;
                                for(i=0;i<11;i++){
                                        if(i!=u&&i!=u2){
                                                au[uc++]=x[i];
                                        }
                                }
                                if(v9(au,result+hh[k]))
                                        return true;
                                if(v9(au,result-hh[k]))
                                        return true;
                                if(result.is_zero()&&hh[k].is_zero())return true;
                                if(!hh[k].is_zero()){
                                        if(v9(au,result*hh[k]))
                                                return true;
                                        if(v9(au,result/hh[k]))
                                                return true;
                                        if(!result.is_zero()){
                                                if(v9(au,lnumber(hh[k])/result))
                                                        return true;
                                        }
                                }
                        }
                }
    }
    return false;
}

bool u9(int x[],const number& r, int size)
{
        int i,j;
        int y[9];
        for(i=0;i<(1<<9)-1;i++){
                if(bc(i)>=size){
                        int c;
                        for(j=0,c=0;j<9;j++){
                                if(i&(1<<j)){
                                        y[c++]=x[j];
                                }
                        }
                        if(vs(y,c,r))
                                return true;
                }
        }
        return false;
```

```
}

bool u10(int x[],const number& r, int size)
{
        int i,j;
        int y[10];
        for(i=0;i<(1<<10)-1;i++){
                if(bc(i)>=size){
                        int c;
                        for(j=0,c=0;j<10;j++){
                                if(i&(1<<j)){
                                        y[c++]=x[j];
                                }
                        }
                        if(c<9){
                                if(vs(y,c,r))
                                        return true;
                        }else if(c==9){
                                if(v9(y,r))
                                        return true;
                        }
                }
        }
        return false;
}

bool u11(int x[], const number& r, int size)
{
        int i,j;
        int y[11];
        for(i=0;i<(1<<11)-1;i++){
                if(bc(i)>=size){
                        int c;
                        for(j=0,c=0;j<11;j++){
                                if(i&(1<<j)){
                                        y[c++]=x[j];
                                }
                        }
                        if(c<9){
                                if(vs(y,c,r))
                                        return true;
                        }else if(c==9){
                                if(v9(y,r))
                                        return true;
                        }else if(c==10){
                                if(v10(y,r))
                                        return true;
                        }
                }
        }
        return false;
}

///Now comes code for N=9
int _tmain(int argc, _TCHAR* argv[])
{
    int x;
    int i,j;
    int u[12]={1,2,3,4,5,6,7,8,9,10,11,12};
    if(v9(u,lnumber(38103))){
```

```
607.          fprintf(stderr,"Fail for test 9\n");
608.      }
609.          fprintf(stderr,"Finish test 9\n");
610.      if(v10(u,lnumber(231051))){
611.          fprintf(stderr,"Fail for test 10\n");
612.      }
613.          fprintf(stderr,"Finish test 10\n");
614.  /*        if(u9(u,number(38103),5)){
615.                fprintf(stderr,"Fail for subdata of 9\n");
616.          }
617.  */        if(u10(u,number(231051),6)){
618.                fprintf(stderr,"Fail for subdata of 10\n");
619.          }
620.          fprintf(stderr,"End of test 10\n");
621.          if(v11(u,lnumber(1765186))){
622.          fprintf(stderr,"Fail for test 11\n");
623.          }
624.          fprintf(stderr,"end for test 11\n");
625.          if(u11(u,number(1765186),7)){
626.                  fprintf(stderr,"Fail for subdata of 11\n");
627.          }
628.          fprintf(stderr,"End of test\n");
629.          return 0;
630.  }
```

OeisWiki <admin@oeis.org> 于2022年6月1日周三 01:07写道：

> Dear Zhao Hui Du, In the OEIS entry https://oeis.org/A060315 there are two links that do not work. Can you tell me what they should be?
> Thank you!
> Neil Sloane, njasloane@gmail.com
>
> --
> This email was sent by N. J. A. Sloane to Zhao Hui Du by the "Email this user" function at OeisWiki. If you reply to this email, your email will be sent directly to the original sender, revealing your email address to them.