

الحمية المشكّلات مع الإرشادي الترميز نقاش

١١١١. على وأن شر بال صي نية مك توبًا كان ال من شور هذا

المشكلة روابط

الأقل على الأرقام من زوج أي بين الهامينغ المسافة تكون بحيث اللفظي الترتيب بحيث من أرقام n أصغر إيجاد تطلب المسألة

حيث رقمًا سيني تج الأرقام من زوج بين 0000 لذلك، $0^0=1, 0^1=1, 0^0=0, 1^1=0$. باستخدام الهماميغ المضافة حساب يمكن
النتيجة. في المتفاعلة الأعداد عدد حساب يمكننا ثم المختلفة. الأرقام المفعلة الأعداد تمثل

10. من أقل على يحتوي الأخير السطري كون أن يمكن بي نما سطر، كل في أرقام 10 الإخراج يكون أن يجب لأنه واحدة مرة خطأ أرتكب جديدة. بصفحة متبوعاً الأخير، في رقم آخر بعد فراغ على يحتوي لدى الأولى الإخراج كان

في الأعلى المستوى مثل يعمل main يجعل مما أفضل، بنيت له أن هي الفائدة الوظيفي. بالبرمجة ممتاز لود هو هذا أن أعتمد
 أخرى. وظيفية برمجة لغات أو `Python`

الفردية. الوظائف تصرح أو مألوفة الغير الوظائف للاختر جديد ملف على أعمل أن لا الطريقة، بهذه
 .

من فصل. بشكّل وتصحيحه وظيفية لكل استخراج يمكن الطريفة، بهذه! يمكن، ما بقدر البرمجة علمتني أيضاً

تختلف لذلک، 111 هو ال أقل. على باتات 2 بتختلف 7 و 0 ل-الثنائية التبعيات كانت إذا ما فحصي عن `hamming(0, 7, 2)` `true`. إلى الوظيفة وتعود بت، 3 ب

من فصل. بشركل ال وظيفه هذه لاخ تبار hamming(0, 7, 2) وإضافة deal() تعلق يمكنني إذن،

□ □ □ □ :

```

/*
{
  ID: lzwjava1
  PROG: hamming
  LANG: C++
}
*/

#include<cstdio>
#include<cstring>
#include<math.h>
#include<stdlib.h>
#include<algorithm>
#include<ctime>

using namespace std;

```

```

const int maxn=1000;

bool hamming(int a,int b,int d)
{
    int c=a^b;
    int cnt=0;
    for(int i=0;i<=30;i++)
    {
        if((1<<i) & c)
        {
            cnt++;
            if(cnt>=d) return true;
        }
    }
    return false;
}

void printArr(int *A,int n)
{
    for(int i=0;i<n;i++)
    {
        printf("%d",A[i]);
        if((i+1)%10==0 || (i==n-1)) printf("\n");
        else printf(" ");
    }
}

bool atLesat(int *A,int cur,int i,int d)
{
    for(int j=0;j<cur;j++)
        if(!hamming(A[j],i,d))
            return false;
    return true;
}

void dfs(int *A,int cur,int n,int d)
{
    if(cur==n)
    {
        printArr(A,n);
    }
}

```

```

        return;
    }

    int st=(cur==0? 0: A[cur-1]+1);
    for(int i=st;;i++)
    {
        if(atLesat(A,cur,i,d))
        {
            A[cur]=i;
            dfs(A,cur+1,n,d);
            return;
        }
    }
}

void deal()
{
    int n,b,d;
    scanf("%d%d%d",&n,&b,&d);
    int A[n];
    dfs(A,0,n,d);
}

int main()
{
    freopen("hamming.in","r",stdin);
    freopen("hamming.out","w",stdout);
    deal();
    //printf("%.2lf\n",(double)clock()/CLOCKS_PER_SEC);
    return 0;
}

/*


```