```
      else pom=(7-(os-d1)+1);
      pom=(pom%7)?pom%7:7;
     }
     else
     {
      for(i=m+1;i<m1;i++)sum+=a[i];
      sum+=(a[m]-d)+1;
      os=sum%7;
      pom=(os+d1)%7;
      if(pom==0)pom=7;
     }
   cout<<"P  V  S  C  P  S  N\n";
   for(i=1;i<=pom-1;i++)cout<<"   ";
   for(i=1;i<=(7-(pom-1));i++)cout<<i<<"  ";
   cout<<endl;
   for(j=i;j<=a[m1];j++)
    if(br!=6)
     {
       if(j>=10){cout<<j<<" ";br++;}
       else {cout<<j<<"  ";br++;}
     }
    else
     if(j>=10){cout<<j<<"\n";br=0;}
     else {cout<<j<<"  "<<"\n";br=0;}
    cout<<endl;
}
```

Задача D2. Милионер

Впечатлен от историята за забогатяване на Хенри Форд (започнал от една ябълка, продал я, купил 2 ябълки и т.н.), която госпожата в училището "Програмистите на България" разказала, Умко, който освен добър ученик е и футболен фен, си изградил следната стратегия за забогатяване, залагайки в „ Еврофутбол":

- За всеки нов тираж се прави един залог с коефициенти 2 или 3. Това значи, че ако заложим К лв. **печалбата** ще бъде съответно 2.К лв или 3.К лв.
- За всеки тираж заложената сума се определя от **печалбата** от предходен тираж увеличена с 1 лв. Всяка **печалба** се залага два пъти: веднъж с коефициент 2 и веднъж с коефициент 3. Спазва се правилото, че новата **печалба** не трябва да е по-малка от предходната.

Помогнете на Умко да намери след колко залагания, следвайки горната стратегия, ще стане милионер, ако в началото заложи 1 лв. Напишете програма **milioner.exe**, която въвежда цяло число ($1<n\leq 440000$) и извежда след най-малко колко залагания ще се получи **печалба** не по-малка от **n** и каква ще е стойността на тази **печалба**.

Вход: 9
Изход: 5 9
Печалбите са съответно: 2 3 6 8 **9**
Вход:25
Изход:11 26
Печалбите са съответно: 2 3 6 8 9 12 14 18 20 21 **26**

Коментар: Задачата се състои в генериране на елементи на множество по даден алгоритъм. Предложеното решение извършва тази генерация итерационно, като се използуват вече генерираните елементи съхранявани в масив и се избира по-малкия от новополучените. Използуват се два брояча за генерационните клонове за запазване на възходящия ред на получаване на елементите. С цел икономия на памет елементите неучастващи в следващите стъпки се изтриват. Стандартното решение ще покрие само част от тестовете.

```
#include <iostream.h>
```