

## 7. Plata unei sume cu bancnote de valori date

Se dau o suma și  $n$  tipuri de bancnote având valori de  $a_1, a_2, \dots, a_n$  lei. Se cer toate modalitățile de plată a sumei utilizând aceste bancnote. Presupunem că dispunem de un număr suficient de bancnote de fiecare tip.

Exemplu: Pentru  $\text{Suma}=5$ ,  $n=3$  (trei tipuri de bancnote) cu valorile 1,2,3, soluțiile sunt:

**Sol 1:** 1 de 2, 1 de 3 **Sol 2:** 1 de 1, 2 de 2 **Sol 3:** 2 de 1, 1 de 3 **Sol 4:** 3 de 1, 1 de 2 **Sol 5:** 5 de 1

Reprezentarea soluției: - soluția are  $n$  componente

-fiecare componentă  $x_k$  are valori în mulțimea  $\{0,1,2,\dots,\text{Suma}/a_k\}$

Condiția de validare: - suma parțială  $s=a[1]*x[1]+a[2]*x[2]+\dots+a[k]*x[k] < \text{Suma}$

-suma parțială  $s$  se actualizează la adăugarea sau eliminarea unei valori  $x[k]$

### Varianta recursivă

```
1. #include<iostream>
2. using namespace std;
3. int x[101],a[101];//x solutia, a valoatile bancnotelor
4. int n,s,Suma,nr;
5.
6. void tipar(int k)
7. {
8.     int i;nr++;
9.     cout<<"Solutia "<<nr<<":"<<endl;
10.    for(i=1;i<=k;i++)
11.    {
12.        if(x[i]>0)
13.            cout<<x[i]<<" de "<<a[i]<<endl;
14.    }
15.    cout<<endl;
16. }
17.
18. void back(int k)
19. {
20.    if(s==Suma)tipar(k-1);
21.    else
22.    {
23.        x[k]=-1;
24.        while(x[k]*a[k]+s<Suma && k<n+1)
25.        {
26.            x[k]=x[k]+1;
27.            s=s+x[k]*a[k];
28.            back(k+1);
29.            s=s-x[k]*a[k];
30.        }
31.    }
32. }
33.
34. int main()
35. {
36.    int i;
37.    cout<<"Suma="; cin>>Suma;
38.    cout<<"n="; cin>>n;
39.    for(i=1;i<=n;i++)
40.    {
41.        cout<<"a["<<i<<"]="";
42.        cin>>a[i];
43.    }
44.    back(1);
45.    return 0;
46. }
```