

Knapsack

a) Pentru rezolvarea pseudo-polinomiala a exercitiului din enunt vom folosi programarea dinamica.

```
int pseudopolynomial(int S[], int K, int n)
{
    if(!n || !K)
    {
        return 0;
    }

    if(S[n-1] > K)
    {
        return pseudopolynomial(S, K, n-1);
    }
    else
    {
        int var = pseudopolynomial(S, K, n-1);
        int var2 = pseudopolynomial(S, K-S[n-1], n-1)+S[n-1];
        return std::max(var, var2);
    }
}
```

b) Notam algoritmul optim cu Opt. Notam algoritmul actual cu Alg

In algoritmul nostrum exista 2 cazuri :

- daca Alg nu intra niciodata pe else, algoritmul Alg = Opt deoarece aduna toate numerele pe care le intalneste.

-daca intra pe else, inseamna ca nu putem aduna numarul la care am ajuns astfel incat sa obtinem $<K$, deci, unul dintre cele 2 numere(s sau x) este automat mai mare decat jumatate din K. Daca numarul intalnit(x) este mai mare decat suma(s), atunci in locul sumei salvam x-ul.

Deci, $Alg > Opt / 2$.

```
int aproximativ(std :: istream &f, int K)
{
    int x;
    int s = 0;
    f >> x;
    while (x != -1)
    {
        if(x <= K-s){
            s+=x;
        }
        else if(x>s)
        {
            s=x;
        }
        f >> x;
    }
    return(s);
}
```