***Näide.***

**ÜLESANNE**

***Sisestada N kolmekohalist täisarvu(arvude 1 number kauba number ja kaks järgist selle hind) ja leida millised kaubad nendest saaks osta R raha eest(mõeldud on ühe kauba ostmine***).

**Lahendus( üks võimalik versioon) siin peaks olema UML, kuid võib olla ka üldine kirjeldus**

1.**Koostame üldise lahenduse =>** arvu N sisestus koos kontrolliga, et positiivne täisarv. Kuna ülesandes on öeldud, et esimene number on kauba number, siis neid ei saa olla rohkem kui 9. **=>** sisestada 3 kohalised arvud arv[N] koos kontrolliga( suuremad 99st ja väiksemad 1000 st) **=>**sisestada arv R**=>**kordusega alustada arv[i ]kontrolli jagades alguses selle 100ga ning võrreldes jääki R-ga, sobivuse korral salvestada massiivi SOB[N].Massiiv võiks olla loodud enne**=>**väljastada SOB[N].***NB! selle skeemi alusel saame jaotada alamülesanneteks***

**2. Koostame main() funktsiooni . . .**

int main ()

{int N,R;

int p=9, p1=99,p2=1000;// suurused sisestatava arvu kontrollimiseks

***N=sisse(’’ kaupade arv’’,p);*** // täisarvude sisestuseks kasutame ühte funktsiooni

***R=sisse(’’rahasumma’’,99);*** // piiri ei pea arvestama aga võib, siis pöördumisel anname väärtuse

int arv[N], SOB[N]; // kuna max N on väike ja on teada, võib kasutada ka 9-t

***loemeka(N,arv,p1,p2);***// alumist ja ülemist piiri tuleb arvestada

***sortim(N,R,arv,SOB);***

***valjTu(SOB);***

return 0;

}

**3. Koostame prototüübid, mis antakse edasi funktsioonide kirjutajale(-tele)**

int sisse(char[],int);

void loemeka(int, int [],int,int);

void sortim(int,int,int[],int[]);

void valjTu(SOB);