RAČUNALNIŠTVO I
1997/98

3. domača naloga

9. april 1998

Kot osnova za vaše delo morajo obvezno biti moduli iz prve, oziroma druge domače naloge, ali tisti, ki smo jih skupaj sestavili na vajah oziroma predavanjih. Te moduli najdete v našem WWW sestavku ali na področju

`Z:\Predavanja\Polde` .

1. V prvi domači nalogi smo naredili izvedbeni modul za večnatančnostno aritmetiko v jeziku c. V drugi domači nalogi smo nato izdelali definicijski modul `Cint` za večnatančnostno aritmetiko v jeziku c++ (razred `Cint`). V tej domači nalogi bomo napisali še izvedbeni modul `Cint` istoimenskega razreda.

   NALOGA: *Napišite izvedbeni in testni modul* `Cint` *za delo z večnatančnostno aritmetiko nad celimi števili poljubne natančnosti.*

   Nad predmeti iz razreda celih števil `Cint` („**C**lass **int**eger“) lahko izvajamo osnovne računske operacije (prištevanje, odštevanje, množenje in deljenje) ter pretvorbe (prirejanja) med različnimi predstavitvami. Poleg notranje predstavitve celih števil, ki jo uporablja razred `Cint`, modul pozna še običajno predstavitev (`long`) in predstavitev z nizom (`char *`). Razred `Cint` je definiran kot:

```
class Cint {
/* Desc:    Razred stevil za vec-natancnostno racunanje. Razred
            omogoca spremembo predstavitve iz long in iz niza (string,
            char *). V predstavitvi kot niz ima obliko (brez
            narekovajev): "<predznak><stevke>". Predznak je lahko "+",
            "-", ali "" (nic). V slednjem primeru se privzame, da je
            predznak "+" -- to je, stevilo je nenegativno. Razred
            omogoca tudi spremembo predstavitve nazaj v niz.

*/

//-------------------------------------------------------------------
//--------------------------------------------[ ZASEBNO, NE SKILI!! ]
private:
  int mPredznak;                // predznak stevila
  int mStevStevk;               // stevilo stevk
```

```
  int * mStevke;                // same stevke; njihovo stevilo se
                                // spreminja po potrebi

//--------------------------------------------------------------------
//-----------------------------------------------------------[ JAVNO ]
public:

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
// - - - - - - - - - - - - - - - - - - - - - -[ ustvarjanje / unicevanje ]
  Cint();
/* Desc:   Konstruktor. Ustvari novo stevilo z vrednostjo nic.
   Parms:  -
   Result:
   Pre:
   Post:   this = 0
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Cint(char * stev);
  Cint(int stev);
/* Desc:   Konstruktor. Ustvari novo stevilo z vrednostjo stev. Ce
           vrednosti stev ne moremo prirediti ima predmet vrednost 0.
   Parms:  stev -- prepisovano stevilo (po vrednosti)
   Result:
   Pre:
   Post:   (this = stev) OR (this = 0)
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  bool Priredi(char * stev);
  bool Priredi(long stev);
  bool Priredi(Cint stev);
/* Desc:   Stevilo, ki ga predstavlja stev, prepisemo v ta predmet. Ce
           je predmet pred tem hranil neko drugo vrednost, slednjo
           pozabimo. Ce prepisa ne moremo izvesti, vrnemo false.
   Parms:  stev -- prepisovano stevilo (po vrednosti)
   Result: RESULT
   Pre:
   Post:   IF (RESULT) THEN this = stev ENDIF
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  ~Cint();
/* Desc:   Destruktor.
   Parms:  -
   Result:
   Pre:
   Post:
*/
```

```
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
// - - - - - - - - - - - - - - - - - - - - - - - [ matematicne operacije ]
  bool Pristej(Cint stev);
/* Desc:   Pristejemo stevilo stev. Ce pristevanja ne moremo izvesti,
           naznacimo to z vrnitvijo vrednosti false.
   Parms:  stev -- pristevatelj (po vrednosti)
   Result: RESULT
           ( Ce rezultat ustreza pogojem, funkcija vrne T, sicer F. )
   Pre:
   Post:   IF RESULT THEN this'= this + stev
           ELSE this' = this ENDIF
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  bool Odstej(Cint stev);
/* Desc:   Odstejemo stevilo stev. Ce odstevanja ne moremo izvesti,
           naznacimo to z vrnitvijo vrednosti false.
   Parms:  stev -- odstevatelj (po vrednosti)
   Result: RESULT
           ( Ker rezultat vedno ustreza pogojem, nam funkcija vrne T. )
   Pre:
   Post:   IF RESULT THEN this'= this - stev
           ELSE this' = this ENDIF
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  bool Zmnozi(Cint stev);
/* Desc:   Stevilo, ki ga predstavlja ta predmet, zmnozimo s stev. Ce
           mnozenja ne moremo izvesti, naznacimo to z vrnitvijo
           vrednosti false.
   Parms:  stev -- faktor (po vrednosti)
   Result: RESULT
           ( Ce rezultat ustreza pogojem, funkcija vrne T, sicer F. )
   Pre:
   Post:   IF RESULT THEN this'= this * stev
           ELSE this' = this ENDIF
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  bool Zdeli(Cint stev, Cint &ostanek);
/* Desc:   Stevilo, ki ga predstavlja ta predmet, zdelimo s stev.
           Kolicnik deljenja se shrani v ta predmet, ostanek pa v
           ostanek. Ce deljenja ne moremo izvesti (recimo stev = 0),
           naznacimo to z vrnitvijo vrednosti false.
   Parms:  stev -- delitelj (po vrednosti)
           ostanek -- ostanek (po referenci)
   Result: RESULT
   Pre:
   Post:   IF RESULT THEN
             (this= this' * stev + ostanek) AND
```

```
            (|stev| < ostanek <= 0)
          ELSE this' = this ENDIF
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - [ poizvedovanja ]
  char * Vrednost();
/* Desc:   Stevilo, ki ga predstavlja ta predmet, vrnemo v
           predstavitvi niza. Uporabnik mora po uporabi sprostiti
           pomnilnik (prostor), v katerem je shranjeno vrnejno
           stevilo. Ce stevilu ne moremo spremeniti predstavitve,
           vrnemo NULL.
   Parms:  -
   Result: RESULT
   Pre:
   Post:   IF (RESULT <> NULL) THEN RESULT = this ENDIF
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  int Primerjaj(Cint stev);
/* Desc:   Stevilo, ki ga predstavlja ta predmet, primerjamo s stev.
           Vrnemo -1, 0 ali +1 (glej spodaj).
   Parms:  stev -- primerjanec (po vrednosti)
   Result: RESULT
   Pre:
   Post:   IF    (this < stev) THEN RESULT = -1
           ELSIF (this = stev) THEN RESULT =  0
           ELSIF (this > stev) THEN RESULT = +1
           ENDIF
*/

}; // Cint
```

Testni modul naj omogoča podobne operacije, kot testna modula v prvi in drugi domači nalogi: seštevanje (`S`), odštevanje (`O`), množenje (`M`) in deljenje (`D`). Pri deljenju naj se količnik shrani v `stev2` in ostanek v `stev1` – `stev1` in `stev2` sta števili, ki ju uporablja testni modul in smo ju srečali že v prejšnjih domačih nalogah. Za razliko od prejšnjih testnih modulov naj ta omogoča dostop do obeh števil (predmetov iz razreda `Cint`). V ta namen sta spremenjena ukaza za vnos in izpis števil. Za vnos ima ukaz sedaj obliko:

`V` < številka >< število >

kjer *številka*, ki ima vrednost *0* ali *1*, pove, v katero od števil se bo shranilo *število*. Ukaz za izpis ima podobno obliko:

`I` < številka >

le da *številka* tu pove, katero od števil se bo izpisalo.

Poleg tega naj testni modul omogoča še primerjanje (ukaz `P`), ki takoj tudi izpiše rezultat primerjanja: *-1*, *0* ali *1*.

Sicer je obnašanje testnega modula podobno prejšnjima:

```
V12
V2-1000
O
I2
```

Izpiše se `-1002`.

2. V drugi domači nalogi smo se ukvarjali z večnatančnostnimi racionalnimi števili in aritmetiko nad njimi v jeziku c, na vajah pa se ukvarjamo s skladom. Sedaj bomo združili oba modula.

   Naloga:

   - *Napišite definicijo razreda racionalnih števil* `Crac` *za večnatančnostno aritmetiko.* Pri tem se zgledujte na razred `Cint` iz prve naloge.
   - *Napišite definicijo razreda skladov racionalnih števil* – t.j. predmetov iz razreda `Crac`, ki ste ga definirali zgoraj. Pri tem se zgledujte na razred `Sklad` z vaj.

Rok za oddajo: *8. 5. 1998. do 19h* v asistentov predal pri vratarju na FMF ali po elektronski pošti.