

OSY Home 1

Deadline: 04.10.2025, 0:00, Time remaining: 5 days, 8 hours

Select multiple files using CTRL or SHIFT or **drag them to this window**

Assignment

Upload ?

1. Nejprve se zkuste jen implementovat postupně dva programy a otestujte si je.
2. Ve druhém kroku si již vyzkoušené programy začněte rozdělovat do adresářů, vytvářet Makefile a knihovny.

I Úkol

Nepoužívat C++ streams! Používejte funkce `fprint/fscanf/fread/fwrite/...` nebo `read/write`.

Čísla bankovních účtů se generují dle pravidla ČNB, tabulka *Váhy pro kontrolní algoritmus*.

V řeči jazyka C jde o pole `vahy[] = { 1, 2, 4, 8, 5, 10, 9, 7, 3, 6 };`.

Zde [ověření](#) korektního čísla.

1. Napište program `gennum S N`, který bude od čísla `S` generovat řadu čísel a vypíše je na `stdout`.

`S` určuje počáteční číslo a v C musí být typu `long`, `N` říká, kolik čísel bude na výstupu.

Pokud bude uvedeno jen `S`, pak `N = 1000`.

Čísla vždy vypisujte na 10 míst i s levými nulami.

Funkci generátoru mějte implementovanou v samostatné funkci.

Test:

```
./gennum 3000030000
./gennum 5000050000 100
./gennum 7000070000 10000 > numbers.txt
```

2. Napište program `verbank [-v]` bude číst čísla ze `stdin` a bude kontrolovat, zda čísla jsou platné bankovní účty.

Program bude čísla vypisovat a za každé napíše informaci, zda je účet platný či neplatný.

`-v` určuje, že se budou vypisovat jen platné účty.

Verifikaci mějte i zde v samostatné funkci.

```
Verifikace: for ( ... ) { sum += ( cislo_uctu % 10 ) * vahy[ i ]; cislo_uctu /=
```



Test:

```
./verbank # read from terminal until <ctrl-d>
./verbank < numbers.txt
./gennum 2000020000 5000 | ./verbank -v
```

3. Přidejte programu argument **-b**, aby vstup i výstup byl v binární podobě, tzn. že v případě prvního programu se funkce **printf** bude nahrazovat např. **write**. Druhý program **scanf/read**.

II Úkol

1. U prvního úkolu přesuňte funkci generující čísla do samostatného souboru **.c** a vytvořte z něj statickou knihovnu. Tu připojte ke svému programu.
2. Program pro ověření správnosti čísla bankovního účtu si přesuňte do nového adresáře a funkci do samostatného zdr. souboru a vytvořte z něj dynamickou knihovnu. Tu připojte ke svému programu a ověřte funkcionalitu obou programů stejně, jako u úkolů I.
3. Vytvořte si v samostatném adresáři druhou dynamickou knihovnu pro ověřování platnosti rodných čísel, viz **MV**. **Funkce i dynamická knihovna se musí jmenovat stejně, jako funkce a knihovna v předchozím bodě.**

Změnou nastavení **export LD_LIBRARY_PATH=sem** a **LD_LIBRARY_PATH=tam** si ověřte, že pro stejnou sadu čísel je možno ověřit desetimístná čísla jako čísla účtů i jako rodná čísla.

Připojení správné knihovny si můžete ověřit už před spuštěním programu pomocí programu **ldd ./mujprogram**.