📄 Nimen lähdegeneraattorin Dokumentaatio

1. 🚀 Yleiskatsaus

Tämä C-kielellä toteutettu ohjelma on tarkoitettu suomalaisten nimiä generoivaan projektiin, joka hyödyntää nimitietoja eri vuosikymmeniltä. Ohjelma lataa kolme erilaista nimilistaa ja antaa käyttäjän valita aikakauden, jonka perusteella nimi (etunimi, keskinimi ja sukunimi) luodaan satunnaisesti.

---

2. 🧱 Datarakenteet (Structs)

Ohjelma hyödyntää kahta päärakennetta nimien tallentamiseen ja käsittelyyn:

Rakenne	Kuvaus	Avainkentät	Käyttötarkoitus
NameList	Yksinkertainen lista nimimerkkijonoille.	char **names (nimitaulukko), int count (nimien määrä)	Käytetään **sukunimien** ja CSV-tiedostojen yksittäisten sarakkeiden tallentamiseen.
DecadeData	Koko CSV-tiedoston säilö, joka käsittelee vuosikymmenittäin jaoteltua dataa.	char **decades (otsikot), NameList *lists (lista NameList-rakenteita), int num_decades (sarakkeiden/vuosikymmenten lkm)	Käytetään **etunimien** ja **keskinimien** tallentamiseen, joissa jokainen sarake edustaa yhtä vuosikymmentä.

---

3. 💾 Nimien Latausfunktiot

Ohjelma sisältää kaksi ensisijaista latausfunktiota eri tiedostomuodoille:

- **load_names_multi_column(const char *filename, DecadeData *data)**
  - **Käyttö:** Ladataan monisarakkeiset CSV-tiedostot, kuten etunimet ja keskinimet.
  - **Toiminta:** Lukee ensin otsikkorivin ja luo jokaiselle otsikolle uuden NameList-rakenteen (DecadeData.lists). Sen jälkeen lukee rivi riviltä ja lisää nimet oikeaan NameList-listaan sarakkeiden perusteella.
- **load_names_simple(const char *filename, NameList *list)**
  - **Käyttö:** Ladataan yksinkertaiset tekstitiedostot (yksi nimi per rivi) tai CSV-tiedoston vain ensimmäinen sarake.
  - **Toiminta:** Lukee tiedoston rivi kerrallaan ja tallentaa jokaisen nimen suoraan annettuun NameList-rakenteeseen. Käytetään pääasiassa **sukunimien** lataamiseen.

---

4. ⚙️ Pääohjelman (main) Logiikka

Vaihe	Toiminto	Tarkoitus
1. Alustus	setlocale(), srand()	Asettaa lokalisoinnin ja satunnaislukugeneraattorin siemenen.
2. Datan Lataus	Kutsut load_names_multi_column ja load_names_simple	Lataa etunimet (DecadeData), keskinimet (DecadeData) ja sukunimet (NameList) muistiin.
3. Tarkistukset	if (first_names.num_decades == 0 ...)	**Kriittinen vaihe:** Tarkistaa, että pakolliset etunimet ja sukunimet on ladattu. Jos ei, ohjelma poistuu virheellä (return 1).
4. Käyttäjän Syöte	print_available_decades(), scanf()	Näyttää käytettävissä olevat vuosikymmenet ja pyytää käyttäjältä valinnan.
5. Nimen Generointi	rand() \% count	Valitsee: **a)** etunimen valitusta vuosikymmenestä, **b)** sukunimen koko sukunimilistasta, ja **c)** keskinimen valitusta vuosikymmenestä 50% todennäköisyydellä.
6. Muistin Vapautus	free_decade_data(), free_names()	Vapauttaa dynaamisesti varatun muistin estäen muistivuodot (memory leaks). **Pakollinen vaihe.**

---

5. Muistinhallinta (Memory Management)

Jokaiselle ladatulle nimelle ja tietorakenteelle varataan muisti dynaamisesti (malloc, realloc, strdup). Siksi on välttämätöntä, että vastaavat vapautusfunktiot kutsutaan ohjelman lopussa (tai virhetilanteessa).

- void free_names(NameList *list)
- void free_decade_data(DecadeData *data)

Nämä funtiot käyvät läpi jokaisen merkkijonon ja vapauttavat sen muistin (free()), vapauttavat osoitinlistat ja lopuksi nollaavat laskurit.