

# Objektno programiranje

## vježba 4

### Uvod u STL

1. Učitati string koji predstavlja rečenicu. Napisati funkciju koja iz stringa izbacuje sve praznine koje se nalaze ispred znakova interpunkcije i dodaje praznine nakon znaka interpunkcije ako nedostaju.

Primjer: Za rečenicu "Ja bih ,ako ikako mogu , ovu recenicu napisala ispravno .",

Ispravna rečenica glasi: "Ja bih, ako ikako mogu, ovu recenicu napisala ispravno."

2. Napisati funkciju koji prevodi engleske rečenice na **pig latin** jezik. Pravila su sljedeća:

(a) ako riječ počinje samoglasnikom dopisuje se hay na kraj riječi,

(b) inače, svi suglasnici s početka riječi prebacuju se na kraj te se na kraju riječi dopisuje ay.

Rečenice treba spremati u vector, a zatim prevesti slučajno odabranu rečenicu iz vektora.

Primjer: "What time is it?" prevodi se kao "atwhay imetay ishay ithay?"

3. Napisati program u kojem korisnik unosi binarne znamenke (0 i 1) i sprema ih u container. Unosu je kraj kad korisnik unese vrijednost koja ne predstavlja binarnu znamenku. Nakon toga napisati funkciju stuffing\_bit koja primjeni **bit stuffing** algoritam na container. Bit stuffing je tehnika koja se koristi u HDLC protokolima za prijenos podataka u kojima je početak i završetak paketa određen sa nizom bitova 01111110. Da se ne bi sadržaj paketa brkao sa stop-bitovima, potrebno je nakon svakog niza od pet jedinica umetnuti jednu nulu.

Primjer: za uneseni niz bitova

001110111111110001111111011

rezultat je

00111011111101100011111011011.

4. Napisati funkciju unstuffing\_bit koja je inverzna funkciji stuffing\_bit iz predhodnog zadatka. Funkcija prima modificirani container i izbacuje umetnute nule kako bi se dobila originalna sekvenca.

5. Koristeći vector napravite implementaciju igre šibica gdje korisnik igra protiv računala. Pravila ove igre su vrlo jednostavna. Pred 2 igrača postavi se 21 šibica. Igrači se izmjenjuju i uklanjaju 1, 2 ili 3 šibice odjednom. Igrač koji je prisiljen uzeti posljednju šibicu gubi.

Korisnik unosi izbor, dok se za odabir računala bira slučajnim izborom. Igraču se mora dati prednost, tako da računalo prvo započinje igru.

Prije pisanja koda promislite o problemu, koji su mogući slučajevi, kad je sigurna pobjeda, kad je neizbježan poraz.