

Vaje pri predmetu Programiranje 2

Teden 14

UTF-8

Naloga

UTF-8 je standard za kodiranje nizov, sestavljenih iz znakov abecede Unicode. Po tem standardu so znaki s kodami od 0 do $2^7 - 1$ zapisani z bajtom oblike *0bbbbbbb*, kjer *bbbbbb* predstavlja dvojiški zapis kode znaka, ki ga z vodilnimi ničlami dopolnimo do dolžine 7, znaki s kodami od 2^7 do $2^{11} - 1$ pa so predstavljeni z zaporedjem bajtov *110bbbb* in *10cccc*, kjer *bbbbcccc* predstavlja dvojiški zapis kode znaka, ki ga z vodilnimi ničlami dopolnimo do dolžine 11.

Na primer, znak **C** ima kodo 67 (*1000011* po dvojiško), zato je zapisan z bajtom *01000011*, znak **Č** pa ima kodo 268 (*00100001100* po dvojiško), zato je zapisan z zaporedjem bajtov *11000100* in *10001100*.

Napišite program, ki na podlagi podane dvojiške datoteke, ki vsebuje zaporedje znakov, kodiranih po standardu UTF-8, izdela dvojiško datoteko s podanim imenom in vanjo po vrsti zapiše kode znakov v obliki števil tipa `short`. Imeni vhodne in izhodne datoteke sta podani kot argumenta ukazne vrstice.

Testni primer 1

V tem testnem primeru program pokličemo z argumentoma `test01.bin` in `test01.bres`. Vhodna datoteka `test01.bin` vsebuje 12 bajtov, ki predstavljajo besedo **kr1žišče** s sledečim znakom za prelom vrstice:

Znak	Koda (desetiško)	Koda (dvojiško)	Bajt
k	107	1101011	01101011
r	114	1110010	01110010
1	49	0110001	00110001
ž	382	00101111110	11000101 10111110
i	105	1101001	01101001
š	353	00101100001	11000101 10100001
č	269	00100001101	11000100 10001101
e	101	1100101	01100101
\n	10	0001010	00001010

Pričakovana vsebina izhodne datoteke `test01.bres` (zapisana je v referenčni datoteki `test01.bout`) je takšna:

Število	Bajt
107	01101011 00000000
114	01110010 00000000
49	00110001 00000000
382	01111110 00000001
105	01101001 00000000
353	01100001 00000001
269	00001101 00000001
101	01100101 00000000
10	00001010 00000000

Cela števila so v pomnilniku in datotekah predstavljena v (neintuitivnem) načinu *little endian*, kar pomeni, da je na začetku zapisan najmanj, na koncu pa najbolj pomemben bajt (na primer, $353 = 01100001_{(2)} + 2^8 \cdot 00000001_{(2)}$). No, če boste pravilno uporabili funkcijo `fwrite`, se vam s to podrobnostjo ne bo treba ukvarjati.

Vhodna datoteka vsebuje kvečjemu $2 \cdot 10^5$ bajtov.

Dvonivojska konstrukcija

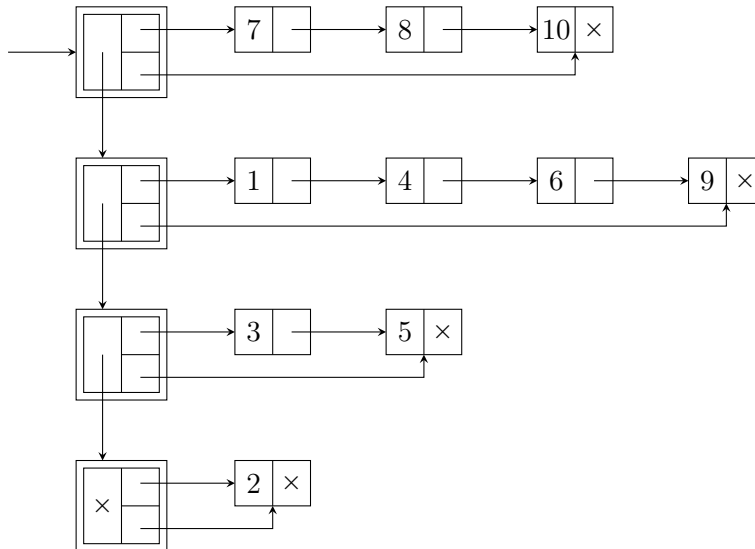
V datoteki `dvonivojskaKonstrukcija.h` sta zapisani deklaraciji, ki omogočata izdelavo konstrukcije, sestavljene iz zunanjega povezanega seznama in množice notranjih povezanih seznamov. Vsako vozlišče zunanjega seznama vsebuje kazalec na začetek in kazalec na konec notranjega seznama:

```
typedef struct Zunanje Zunanje;
typedef struct Notranje Notranje;

struct Zunanje {          // vozlišče zunanjega seznama
    Notranje* prvo;       // prvo vozlišče pripadajočega notranjega seznama
    Notranje* zadnje;     // zadnje vozlišče pripadajočega notranjega seznama
    Zunanje* dol;         // naslednje vozlišče zunanjega seznama (NULL, če ga ni)
};

struct Notranje {         // vozlišče notranjega seznama
    int podatek;          // podatek, ki ga hrani vozlišče (element seznama)
    Notranje* desno;      // naslednje vozlišče (NULL, če ga ni)
};
```

Sledeča slika prikazuje primer dvonivojske konstrukcije. Vozlišča zunanjega seznama (tj. vozlišča tipa `Zunanje`) so dvojno uokvirjena, znak \times pa predstavlja vrednost `NULL`.



V datoteki `dvonivojskaKonstrukcija.c` dopolnite funkcijo

`Zunanje* izdelajDvonivojsko(int* t, int n),`

tako da bo iz elementov tabele dolžine n ($0 \leq n \leq 1000$), ki se prične na naslovu `t`, po sledečih pravilih zgradila dvonivojsko konstrukcijo:

- Pričnite s prazno konstrukcijo. Ta ne vsebuje nobenega vozlišča.
- Naj bo x trenutni element tabele. Poiščite prvi notranji seznam, v katerem je zadnji element manjši ali enak elementu x , in element x dodajte na konec tega seznama. Če tak seznam ne obstaja, dodajte na konec zunanjega seznama nov notranji seznam z elementom x .

Konstrukcijo na gornji sliki dobimo s testnim programom `test01.c`, ki funkcijo `izdelajDvonivojsko` pokliče s tabelo z elementi 7, 1, 4, 3, 2, 6, 5, 8, 10, 9.

Funkcija `izdelajDvonivojsko` naj vrne kazalec na prvo vozlišče zunanjega seznama oziroma `NULL`, če je izhodna konstrukcija prazna.