

Rešitve pisnega izpita (programska jezika Java in C) z dne 1. septembra 2004

1. naloga (25%)

```
public int indeksZadnji(String niz, char znak) {
    for(int i=niz.length()-1; i>=0; i--)
        if( niz.charAt(i) == znak )
            return i;
    return i;
}
```

2. naloga (25%)

```
public class Nakljucje implements Runnable {
    private final int MODUL = 2004;
    private int stevec = 0;
    private boolean konec = false;
    private Thread nit;

    public Nakljucje() {
        if(nit == null) {
            nit = new Thread(this);
            nit.start();
        }
    }

    public void run() {
        while(!konec)
            stevec = (++stevec)%MODUL;
    }

    public int vrniStevilo() {
        return stevec;
    }

    public void konec() {
        konec = true;
    }
}
```

3. naloga (25%)

Datoteko moramo prebrati in popravljeno vsebino zapisati nazaj v isto datoteko. Zato si moramo vsebino datoteke ob branju zapomniti - shranimo jo v polje `tabela`. Velikost datoteke (Število znakov v njej) je poljubna, zato moramo najprej ugotoviti dejansko velikost datoteke, da lahko ustrezno določimo velikost polja za shranjevanje prebranih znakov. To naredimo tako, da datoteko odpremo in preštejemo znake v njej. Ko poznamo velikost datoteke (spremenljivka `dolzina`), lahko rezerviramo ustrezno velik prostor za vse znake iz datoteke.

```
#include <stdio.h>

main(int argc, char *argv[]) {
    FILE *fp;
    int znak = 0;
    char *tabela;
    int i = 0;
    int dolzina = 0;

    fp = fopen(argv[1], "r");
    while( (znak=getc(fp)) != EOF ) {
        dolzina++;
    }
    fclose(fp);
    tabela = (char *)malloc(dolzina*sizeof(char));
    fp = fopen(argv[1], "r");
    while( (znak=getc(fp)) != EOF ) {
        tabela[i] = znak;
        i++;
    }
    fclose(fp);
    fp = fopen(argv[1], "w");
    for( i=0; i<dolzina; i++ ) {
        if( !( tabela[i] == ' ' ) && (i<dolzina-1) && ((tabela[i+1] == 'c') || (tabela[i+1] == 's') || (tabela[i+1] == 'z')) ) )
            fprintf(fp, "%c", tabela[i]);
    }
    fclose(fp);
}
```

Odpiranje in branje datoteke samo za to, da poiščemo število znakov v njej, je nekoliko nerodno. To lahko naredimo tudi s pomočjo ukazov `fseek` in `ftell`, kot je to prikazano v spodnji rešitvi. Del kode, ki se razlikuje, je označen z rdečo:

```
#include <stdio.h>

main(int argc, char *argv[]) {
    FILE *fp;
    int znak = 0;
    char *tabela;
    int i = 0;
    int dolzina = 0;

    fp = fopen(argv[1], "r");
    fseek(fp,0,SEEK_END); // postavimo se na konec datoteke
    dolzina = ftell(fp); // pogledamo dolzino datoteke
    fseek(fp,0,SEEK_SET); // postavimo se spet na zacetek datoteke
```

```
tabela = (char *)malloc(dolzina*sizeof(char));
while( (znak=getc(fp)) != EOF ) {
    tabela[i] = znak;
    i++;
}
fclose(fp);
fp = fopen(argv[1], "w");
for( i=0; i<dolzina; i++ ) {
    if( !( (tabela[i] == '') && (i<dolzina-1) && ((tabela[i+1] == 'c') || (tabela[i+1] == 's') || (tabela[i+1] == 'z')) ) )
        fprintf(fp, "%c", tabela[i]);
}
fclose(fp);
}
```

4. naloga (25%)

```
struct elem {
    double stevilo;
    struct elem *nasl;
};

void duplikati(struct elem *zac) {
    struct elem *p, *q, *t;

    p = zac;
    while(p != NULL) {
        q = p;
        while(q->nasl != NULL) {
            if(q->nasl->stevilo == p->stevilo) {
                t = q->nasl;
                q->nasl = q->nasl->nasl;
                free(t);
            }
            else
                q = q->nasl;
        }
        p = p->nasl;
    }
}
```