

JMBAG		Prezime, ime	
-------	--	--------------	--

Uvod u programiranje

Završni ispit - 27. siječnja 2020.

Rješenja zadataka 1 - 4 napisati na vlastitim papirima i predati ih u košuljici.

Rješenja zadataka 5 - 6 napisati na za to predviđeno mjesto uz tekst zadatka te list sa zadacima predati u košuljici.

1. (10 bodova)

Svaki zapis *tekstne* datoteke `recenice.txt` opisuje jednu rečenicu. Zapis sadrži cjelobrojnu vrijednost n ($n \geq 2$) koja predstavlja broj riječi u rečenici, nakon čega slijedi n cijelih brojeva (*int*) koji predstavljaju šifre riječi od kojih je rečenica sastavljena. Cijeli brojevi unutar jednog zapisa datoteke međusobno su odijeljeni zarezima.

Svaki zapis *binarne* datoteke `rijeci.bin` sadrži šifru riječi (*int*) i riječ. Riječ je niz znakova (*string*) najveće moguće duljine 20+1 znak. Redni broj zapisa u datoteci `rijeci.bin` odgovara šifri riječi.

Napisati program kojim će se na zaslon ispisati riječi rečenica iz `recenice.txt`, ali tako da se ispišu samo one riječi koje sadrže barem jedan **mali** samoglasnik (a, e, i, o, u). Svaku rečenicu treba ispisati u svom retku.

Primjer sadržaja `recenice.txt`

Primjer ispisa na zaslon

```
4,626,11247,32,11250↵
2,31,30↵
3,628,33,629↵
2,626,11252↵
3,11249,11251,627↵
5,11253,33,630,29,34↵
...
```

```
radu·je·spas·↵
Prazan·↵
nije·voda·↵
↵
Travu·kosi·↵
Nitko·nije·za·to·kriv·↵
...
```

Primjer sadržaja
`rijeci.bin`

...	...
29	to
30	vrt
31	Prazan
32	je
33	nije
34	kriv
...	...
626	U
627	srp
628	Krv
629	voda
630	za
...	...
11247	radu
11248	su
11249	Travu
11250	spas
11251	kosi
11252	NOMENKLATURI
11253	Nitko
...	...

2. (10 bodova)

Napisati funkciju `traziZnak` koja kao parametre prima cijeli broj n (*int*) i nizove znakova (*string*) `niz1` i `niz2`. Funkcija treba u nizu `niz1` pronaći prvi znak koji se barem n puta pojavljuje u nizu `niz2` te u pozivajući program preko imena funkcije vratiti pokazivač na taj znak. **Napomena:** u funkciji se ne smiju koristiti pomoćna polja.

U nastavku napisati funkciju `main` ("glavni program") u kojoj će se pomoću funkcije `fgets` s tipkovnice učitati jedan niz znakova koji sigurno nije dulji od 40+1 znakova, a zatim pomoću funkcije `traziZnak` u njemu pronaći prvi znak koji se barem 3 puta pojavljuje u nizu "BbaacacBacBc". Na zaslon ispisati ili pronađeni znak ili poruku "nema takvog znaka".

Primjeri izvršavanja programa

```
Upisite niz>>·bijekcija↵
Rjesenje: c
```

```
Upisite niz>>·The bottle↵
Rjesenje: nema takvog znaka
```

3. (10 bodova)

Napisati funkciju `gen` koja pri svakom pozivu vraća dvije vrijednosti: 1) realni broj (*double*) iz zatvorenog intervala $[1, 10]$ koji je generiran pomoću generatora pseudo-slučajnih brojeva; 2) logičku vrijednost *istina* ako je generirani broj veći od broja koji je bio generiran pri prethodnom pozivu funkcije tijekom izvršavanja programa, inače logičku vrijednost *laž*.

Generator pseudo-slučajnih brojeva treba inicijalizirati u funkciji `gen`.

Primjer vrijednosti koje funkcija `gen` vraća pri uzastopnim pozivima

```
9.3966, false
1.1695, false
1.7358, true
4.2952, true
3.5950, false
2.0034, false
6.6353, true
```

4. (10 bodova)

Napisati funkciju `brojiBitove` koja za zadani cijeli broj tipa *unsigned int* vraća dvije vrijednosti: 1) koliko bitova zadanog broja je postavljeno na jedinicu; 2) u koliko bajtova zadanog broja su svi bitovi postavljeni na jedinicu. Može se pretpostaviti da je podatak tipa *unsigned int* pohranjen u 4 bajta.

Primjeri:

- za zadani broj 25 funkcija treba vratiti cijele brojeve 3, 0
 - jer je $25_{10} = 00000000\ 00000000\ 00000000\ 00011001_2$
- za zadani broj 16720895 funkcija treba vratiti cijele brojeve 19, 2
 - jer je $16720895_{10} = 00000000\ 11111111\ 00100011\ 11111111_2$.

5. (2 boda)

Zadan je sljedeći programski odsječak:

```
for (i = poc; i <= t; i = i + k) {
    x = x + i * t;
}
```

Programski odsječak realizirati petljom s ispitivanjem uvjeta na kraju (*do-while*). Pri tome nije dopušteno koristiti naredbe `continue`, `break` i `goto`, niti je dopušteno definirati dodatne varijable.

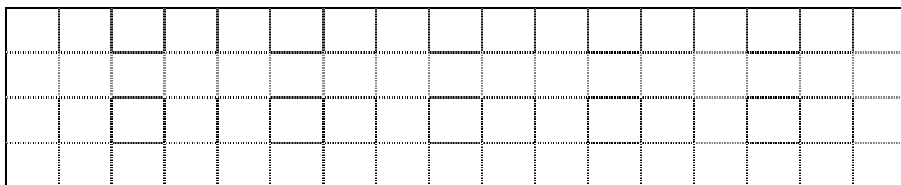
prostor predviđen za rješenje 5. zadatka

6. (3 boda)

Prikazati što će se tijekom izvršavanja odsječka programa ispisati na zaslonu, vodeći računa o točnoj poziciji ispisanih znakova: svaki "kvadratić" na papiru predstavlja mjesto na zaslonu na koje se ispisuje po jedan znak; svaki "redak kvadratića" predstavlja jedan redak na zaslonu. Pretpostaviti da je preko tipkovnice upisan sadržaj prikazan u desnom okviru.

```
int a = 1, b = 2, c = 3, d = 4;
int polje[5] = {5, 6, 7}, *p = polje;
a = scanf("%d%d %d", &b, &c, &d);
printf("%d %d %d %d\n", a, b, c, d);
printf("%d %d %d", *p, *(p + 3), *p + 3);
```

..6.1B2.┘



Rješenja:

```
1. #include <stdio.h>
#include <string.h>

int main(void) {
    struct rijec_s {
        int sifRijec;
        char rijec[20 + 1];
    } rijecZapis;

    FILE *txt = fopen("recenice.txt", "r");
    FILE *bin = fopen("rijeci.bin", "rb");

    int brojRijeci, sifRijeci;
    while (fscanf(txt, "%d", &brojRijeci) == 1) {
        for (int i = 0; i < brojRijeci; ++i) {
            fscanf(txt, "%d", &sifRijeci);
            fseek(bin, (long)(sifRijeci - 1) * sizeof(rijecZapis), SEEK_SET);
            fread(&rijecZapis, sizeof(rijecZapis), 1, bin);
            if (strpbrk(rijecZapis.rijec, "aeiou") != NULL) {
                printf("%s ", rijecZapis.rijec);
            }
        }
        printf("\n");
    }

    fclose(txt);
    fclose(bin);

    return 0;
}

2. #include <stdio.h>
#include <string.h>
char *traziZnak(int n, char *niz1, char *niz2) {
    for (int i = 0; i < strlen(niz1); ++i) {
        int brojPojava = 0;
        for (int j = 0; j < strlen(niz2); ++j) {
            if (*(niz1 + i) == *(niz2 + j)) {
                if (++brojPojava == n) {
                    return niz1 + i;
                }
            }
        }
    }
    return NULL;
}

int main(void) {
    char nizUlaz[40 + 1], *pc;

    printf("upisite niz > ");
    fgets(nizUlaz, 40 + 1, stdin);

    pc = traziZnak(3, nizUlaz, "BbcaaacBacBc");
    printf("Rjesenje: ");
    if (pc == NULL) {
        printf("nema takvog znaka");
    } else {
        printf("%c", *pc);
    }
    return 0;
}
```

```
3. #include <stdlib.h>
#include <time.h>
#include <stdbool.h>
```

```
void gen(double *broj, bool *jestVeci) {
    static int prviPoziv = true;
    static double preth;
    if (prviPoziv) {
        srand((unsigned int)time(NULL));
    }
    *broj = (double)rand() / RAND_MAX * (10 - 1) + 1;
    if (prviPoziv) {
        *jestVeci = false;
        prviPoziv = false;
    } else {
        *jestVeci = *broj > preth;
    }
    preth = *broj;
    return;
}
```

```
/* Ovo dalje nije dio rjesenja, služi samo za testiranje */
#include <stdio.h>
int main(void) {
    double broj;
    bool jestVeci;
    for (int i = 1; i <= 7; ++i) {
        gen(&broj, &jestVeci);
        printf("%.4lf,   %s\n", broj, jestVeci ? "true" : "false");
    }
    return 0;
}
```

```
4. void brojiBitove(unsigned int broj, int *brojBitova, int *brojBajtova) {
    *brojBitova = *brojBajtova = 0;
    for (int i = 0; i < sizeof(unsigned int) * 8; ++i) {
        if ((broj >> i & 0x1) == 1) ++*brojBitova;
    }
    for (int i = 0; i < sizeof(unsigned int); ++i) {
        if ((broj >> i * 8 & 0xFF) == 0xFF) ++*brojBajtova;
    }
}
```

```
/* Ovo dalje nije dio rjesenja, služi samo za testiranje */
#include <stdio.h>
int main(void) {
    unsigned int b;
    int rez1, rez2;
    scanf("%u", &b);
    brojiBitove(b, &rez1, &rez2);
    printf("%u -> %d, %d\n", b, rez1, rez2);
    return 0;
}
```

