

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

Uvod u programiranje

Završni ispit - 25. siječnja 2021.

Napomene:

- Programska rješenja moraju biti usklađena sa stilom pisanja programa koji je propisan na predmetu
- 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. (12 bodova)

Svaki zapis *binarne* datoteke `potrosac.bin` sadrži šifru (*int*) te ime i prezime (niz znakova 30+1 znak) potrošača električne energije. Redni broj zapisa u datoteci `potrosac.bin` odgovara šifri potrošača. U datoteci postoje i "prazni" zapisi (npr. ne postoji potrošač sa šifrom 1).

Svaki zapis *tekstne* datoteke `potrosnja.txt` sadrži šifru potrošača, datum i količinu električne energije izraženu u kWh (realni broj) koju je potrošač potrošio tog datuma.

Napisati program koji će na zaslon ispisati podatke o potrošnji tijekom 7. i 8. mjeseca 2020. godine za potrošače čije ime počinje slovom J. Za svaki zapis o potrošnji koji zadovoljava navedene uvjete, na zaslon ispisati šifru potrošača, količinu potrošene energije, datum potrošnje te ime i prezime potrošača.

potrosac.bin

2	Jan Kolar
5	Ana Jedvaj
6	Jure Zovko
8	Jakov Gould
10	Velimir Turk
11	Jadranka Ban

potrosnja.txt

11	4.8.2020.	80.4
6	12.8.2020.	45.2
6	21.8.2020.	104.8
6	9.2.2019.	100.1
11	27.8.2020.	35.7
10	7.8.2019.	254.1
6	27.9.2020.	98.6
2	12.8.2019.	315.8
5	27.7.2020.	198.1
8	9.7.2020.	9.3

Primjer izvršavanja

...	11	...	80.4	kWh	...	4.8.2020.	...	Jadranka Ban
...	6	...	45.2	kWh	...	12.8.2020.	...	Jure Zovko
...	6	...	104.8	kWh	...	21.8.2020.	...	Jure Zovko
...	11	...	35.7	kWh	...	27.8.2020.	...	Jadranka Ban
...	8	...	9.3	kWh	...	9.7.2020.	...	Jakov Gould

2. (12 bodova)

Napisati funkciju `ispisPodnizova` koja kao parametre prima niz znakova (*string*) i pokazivač na jedan od znakova u tom nizu. **Funkcija** na zaslon u prvom retku ispisuje sve znakove zadanog niza počevši od znaka na kojeg pokazuje zadani pokazivač do **početka** niza, a u sljedećem retku sve znakove počevši od kraja zadanog niza do znaka na kojeg pokazuje zadani pokazivač. **U rješenju nije dopušteno koristiti funkcije iz <string.h>.**

Primjer izvršavanja

tam	invizarbA
lajiret	

U nastavku napisati funkciju `main` ("glavni program") koja će funkciju `ispisPodnizova` pozvati sa sljedećim argumentima: konstantni znakovni niz "**Abrazivni materijal**" i pokazivač na znak **t** u tom konstantnom nizu. Očekuje se da će **funkcija** za tako zadane parametre na zaslon ispisati sadržaj prikazan u primjeru.

3. (12 bodova)

Napisati funkciju `genTocka`. Funkcija pri svakom pozivu generira slučajno odabrane vrijednosti koordinata točke u Kartezijevom koordinatnom sustavu, pri čemu koordinate točke x i y trebaju biti realni brojevi iz zatvorenog intervala $[0.0, 10.0]$. Funkcija u pozivajući program treba vratiti udaljenost generirane točke do točke koja je generirana u prethodnom pozivu funkcije i **strukturu** koja sadrži x i y koordinate generirane točke. Za točku generiranu pri prvom pozivu funkcije, udaljenost do prethodno generirane točke se izračunava kao udaljenost do ishodišta koordinatnog sustava. Generator pseudo-slučajnih brojeva treba inicijalizirati u pozivajućem programu, prije prvog poziva funkcije `genTocka`.

Primjer izvršavanja

..	0.85	...	2.98	...	3.10
..	5.55	...	9.41	...	7.96
..	2.86	...	3.32	...	6.66
..	6.83	...	4.67	...	4.19
..	1.93	...	5.19	...	4.93
..	3.10	...	2.72	...	2.74

U nastavku napisati funkciju `main` ("glavni program") koja će ispisati rezultate 6 uzastopnih poziva funkcije.

Objašnjenje primjera: pri prvom pozivu funkcija je generirala koordinate točke 0.85, 2.98, te u pozivajući program vratila udaljenost točke do ishodišta (3.10) i strukturu koja sadrži upravo generirane koordinate. Pri drugom pozivu funkcija je generirala koordinate točke 5.55, 9.41, te u pozivajući program vratila udaljenost generirane točke do točke s koordinatama 0.85, 2.98 i strukturu koja sadrži upravo generirane koordinate.

4. (10 bodova)

Napisati funkciju koja u pozivajući program vraća indeks retka u kojem se nalazi najmanji član zadanog dvodimenzijskog cjelobrojnog polja s m redaka i n stupaca. Ako više članova polja ima istu najmanju vrijednost, funkcija treba vratiti indeks prvog od redaka u kojem se nalazi član s tom vrijednošću. U funkciji nije dopušteno korištenje pomoćnih polja.

5. (3 boda)

Na predviđena mjesta napisati vrijednost i tip rezultata sljedećeg izraza:

$(5 \mid 11 \ll 2 \ \& \ 14) + 5 * 3. / 2$

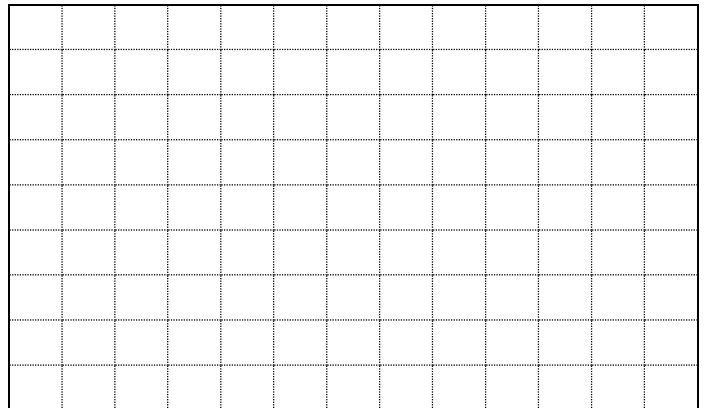
vrijednost	tip
------------	-----

6. (4 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 donjem okviru.

```
int a = 10, b = 10, c = 10, d = 10;
d = scanf("%d%d%d", &a, &b, &c);
printf("%6d%d\n", a, b);
printf("%d%c%d\n", d, c, c);
char niz[] = "ABEF32", *p;
p = &niz[3];
printf("%d%c\n", *(p + 2), *p + 2);
printf("%d\n", *p - *niz);
```

```
-451.....24-↵
22.C1↵
```



Rješenja

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

int main(void) {
    struct {
        int sifPotr;
        char imePrez[30+1];
    } potrosac;

    FILE *fpotrosac = fopen("potrosac.bin", "rb");
    FILE *fpotrosnja = fopen("potrosnja.txt", "r");

    int sifPotr, dan, mj, god;
    float kwh;
    while (fscanf(fpotrosnja, "%d%d.%d.%d.%f", &sifPotr, &dan, &mj, &god, &kwh) == 5) {
        fseek(fpotrosac, (long)(sifPotr - 1) * sizeof(potrosac), SEEK_SET);
        fread(&potrosac, sizeof(potrosac), 1, fpotrosac);
        if (potrosac.imePrez[0] == 'J' && god == 2020 && (mj == 7 || mj == 8)) {
            printf("%6d %7.1f kWh %2d.%d.%d. %s\n",
                potrosac.sifPotr, kwh, dan, mj, god, potrosac.imePrez);
        }
    }
    fclose(fpotrosac);
    fclose(fpotrosnja);
    return 0;
}
```

```
2.    #include <stdio.h>

void ispisPodnizova(char *niz, char *granica) {
    char *p;
    for (p = granica; p >= niz; --p) {
        printf("%c", *p);
    }
    printf("\n");
    for (p = niz; *p != '\0'; ++p);
    for (p = p - 1; p >= granica; --p) {
        printf("%c", *p);
    }
    return;
}

int main(void) {
    char *konst = "Abrazivni materijal";
    ispisPodnizova(konst, konst + 12);
    return 0;
}
```

```

3. #include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

typedef struct {
    double x;
    double y;
} tocka_t;

double genTocka(tocka_t *genT) {
    static tocka_t prethT; // inicijalizira se na {0.,0.}
    double udalj;
    genT->x = (double)rand() / RAND_MAX * 10.;
    genT->y = (double)rand() / RAND_MAX * 10.;
    udalj = sqrt(pow(prethT.x - genT->x, 2.) + pow(prethT.y - genT->y, 2.));
    prethT = *genT;
    return udalj;
}

int main(void) {
    tocka_t tocka;
    srand((unsigned int)time(NULL));
    for (int i = 0; i < 6; ++i) {
        double udalj = genTocka(&tocka);
        printf("%6.2f %6.2f %8.2f\n", tocka.x, tocka.y, udalj);
    }
    return 0;
}

Jednako vrijedna varijanta (odsječak):
tocka_t genTocka(double *udalj) {
    static tocka_t prethT; // inicijalizira se na {0.,0.}
    tocka_t genTocka;
    genTocka.x = (double)rand() / RAND_MAX * 10.;
    genTocka.y = (double)rand() / RAND_MAX * 10.;
    *udalj = sqrt(pow(prethT.x - genTocka.x, 2.) + pow(prethT.y - genTocka.y, 2.));
    prethT = genTocka;
    return genTocka;
}

int main(void) {
    double udalj;
    srand((unsigned int)time(NULL));
    for (int i = 0; i < 6; ++i) {
        tocka_t tocka= genTocka(&udalj);
    }

Jednako vrijedna varijanta (odsječak):
void genTocka(tocka_t *genT, double *udalj) {
    static tocka_t prethT; // inicijalizira se na {0.,0.}
    genT->x = (double)rand() / RAND_MAX * 10.;
    genT->y = (double)rand() / RAND_MAX * 10.;
    *udalj = sqrt(pow(prethT.x - genT->x, 2.) + pow(prethT.y - genT->y, 2.));
    prethT = *genT;
    return;
}

int main(void) {
    tocka_t tocka;
    double udalj;
    srand((unsigned int)time(NULL));
    for (int i = 0; i < 6; ++i) {
        genTocka(&tocka, &udalj);
    }
}

```

5. 20.5 double

[illegible]