

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

## Uvod u programiranje

### Međuispit - 18. studenog 2019.

#### Napomene:

- Programska rješenja moraju biti u skladu 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. (9 bodova)

S tipkovnice učitati dva niza znakova (*string*), nizA i nizB, pri čemu nizovi sigurno neće biti dulji od 40 znakova (uključujući eventualni znak za novi red). U novi niz znakova (*string*) izlaz redom prepisati znakove niza nizA koji se **tri ili više puta** pojavljuju u nizu nizB. Na kraju, niz izlaz ispisati na zaslon pomoću konverzijske specifikacije %s. **Napomena:** dopušteno je koristiti samo funkcije iz stdio.h, a agregatni tipovi podataka smiju se koristiti isključivo za pohranu nizova nizA, nizB i izlaz.

Primjeri izvršavanja programa

|                                   |                    |           |
|-----------------------------------|--------------------|-----------|
| niz A>>PremaJupiteru, pitam!?\n   | niz A>>Pet\n       | niz A>>\n |
| niz B>>PetiPlanetJutrosPluta!!!\n | niz B>>pETpETpET\n | niz B>>\n |
| izlaz>>P.t.t!\n                   | izlaz>>\n          | izlaz>>\n |

#### 2. (9 bodova)

S tipkovnice učitati broj točaka *n*, a zatim *x* i *y* koordinate za *n* točaka. Ispisati koordinate točke (ili točaka) koja je (ili koje su) najbliže ishodištu koordinatnog sustava (*x*=0.0, *y*=0.0).

Primjer izvršavanja programa

|                                |
|--------------------------------|
| Upisati.n>>7\n                 |
| •1.točka>>0.512.35\n           |
| •2.točka>>2.25-2.0\n           |
| •3.točka>>165.10.5\n           |
| •4.točka>>3.02.0\n             |
| •5.točka>>2.02.25\n            |
| •6.točka>>2.25-2.0\n           |
| •7.točka>>2.25-2.1\n           |
| Najblize ishodistu su tocke:\n |
| •2.25-2.00\n                   |
| •2.00-2.25\n                   |
| •2.25-2.00\n                   |

Udaljenost između točaka (*x*<sub>1</sub>, *y*<sub>1</sub>) i (*x*<sub>2</sub>, *y*<sub>2</sub>):

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

#### 3. (9 bodova)

S tipkovnice učitavati cijele brojeve dok god se ne upiše broj koji nije iz skupa prirodnih brojeva. Zatim na zaslon redom ispisati cijele brojeve iz intervala [2, 20] s kojima je djeljivo barem 50% učitanih prirodnih brojeva. Cijeli brojevi koji se učitavaju s tipkovnice ne smiju se pohranjivati u polje jer nije unaprijed poznato koliko bi ih moglo biti učitano.

Primjer izvršavanja programa

|                     |
|---------------------|
| Upisati.broj>>5\n   |
| Upisati.broj>>200\n |
| Upisati.broj>>80\n  |
| Upisati.broj>>20\n  |
| Upisati.broj>>80\n  |
| Upisati.broj>>120\n |
| Upisati.broj>>10\n  |
| Upisati.broj>>6\n   |
| Upisati.broj>>7\n   |
| Upisati.broj>>0\n   |
| Rezultat:2451020    |

**Objašnjenje:** kao rezultat su se ispisali brojevi **2, 4, 5, 10 i 20** jer je 7 od ukupno 9 učitanih prirodnih brojeva djeljivo s **2**; 5 ih je djeljivo s **4**; 7 ih je djeljivo s **5**; 6 ih je djeljivo s **10** i 5 ih je djeljivo s **20**.

#### 4. (3 boda)

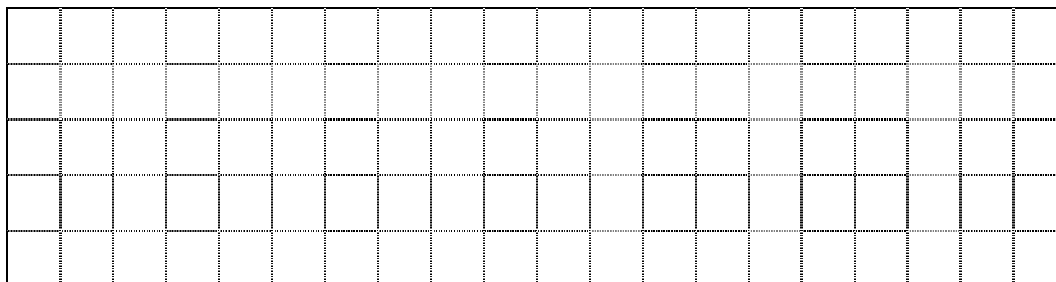
Sljedeći programski odsječak realizirati bez petlje for i naredbi continue, break i goto. Dopušteno je definirati i koristiti najviše jednu dodatnu varijablu.

```
for (i = poc; i <= m; i = i + n) {
    if (i > k) {
        x = x * 1.1f;
        break;
    }
    if (i < t) {
        y = x + y * 0.9f;
    }
    z = y + z / 1.2f;
}
```

#### 5. (3 boda)

Prikazati što će se na zaslonu ispisati tijekom izvršavanja odsječka programa, 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.

```
double x;
unsigned int i = 4U;
i << 2;
printf("%u\n", i);
x = (int) 12.4+8.4 / 2*2;
printf("%.2lf", x);
printf("\n%c%5d", '9' - '0' + 1, 9 - '0' + 1);
```



#### 6. (2 boda)

U skladu sa standardom IEEE 754 za prikaz brojeva standardne preciznosti, u registru je pohranjen negativni realni broj -5.6875. Sadržaj tog registra prikažite u **oktalnom** brojevnom sustavu.

Prostor za odgovor

## Rješenja

```
1. #include <stdio.h>
#define MAXNIZ 40
#define POTREBNO 3

int main(void) {
    char nizA[MAXNIZ + 1], nizB[MAXNIZ + 1], izlaz[MAXNIZ + 1];

    printf("niz A > ");
    fgets(nizA, MAXNIZ + 1, stdin);
    printf("niz B > ");
    fgets(nizB, MAXNIZ + 1, stdin);

    int indNizA = 0, indIzlaz = 0;
    while (nizA[indNizA] != '\0') {
        int brojPojava = 0, indNizB = 0;
        while (brojPojava < POTREBNO && nizB[indNizB] != '\0') {
            if (nizA[indNizA] == nizB[indNizB]) {
                ++brojPojava;
                if (brojPojava == POTREBNO) {
                    izlaz[indIzlaz++] = nizA[indNizA];
                }
            }
            ++indNizB;
        }
        ++indNizA;
    }
    izlaz[indIzlaz] = '\0';
    printf("izlaz > %s\n", izlaz);

    return 0;
}

ili

while (nizB[indNizB] != '\0') {
    if (nizA[indNizA] == nizB[indNizB]) {
        ++brojPojava;
        if (brojPojava == POTREBNO) {
            izlaz[indIzlaz++] = nizA[indNizA];
            break;
        }
    }
    ++indNizB;
}
```

```

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

int main(void) {
    struct tocka_s {
        float x;
        float y;
    };
    int n, i;
    float min_udalj, udalj;
    printf("Upisati n > ");
    scanf("%d", &n);
    struct tocka_s tocke[n];

    for (i = 0; i < n; ++i) {
        printf("%3d. tocka > ", i + 1);
        scanf("%f %f", &tocke[i].x, &tocke[i].y);
        udalj = sqrt(pow(tocke[i].x, 2.) + pow(tocke[i].y, 2.));
        if (i == 0) {
            min_udalj = udalj;
        } else if (udalj < min_udalj) {
            min_udalj = udalj;
        }
    }

    printf("Najblize ishodistu su tocke:\n");
    for (i = 0; i < n; ++i) {
        udalj = sqrt(pow(tocke[i].x, 2.) + pow(tocke[i].y, 2.));
        if (udalj == min_udalj) {
            printf("%.2f %.2f\n", tocke[i].x, tocke[i].y);
        }
    }
    return 0;
}

```

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

#define GG 20
int main(void) {
    int broj;
    int brojac[GG - 1] = {0};
    int i, ucitanih = 0;
    do {
        printf("Upisati broj > ");
        scanf("%d", &broj);
        if (broj > 0) {
            ucitanih++;
            for (i = 2; i <= GG; ++i) {
                if (broj % i == 0) {
                    ++brojac[i - 2];
                }
            }
        }
    } while (broj > 0);
    printf("Rezultat:");
    for (i = 2; i <= GG; ++i) {
        if (brojac[i - 2] >= (ucitanih + 1) / 2) {
            printf(" %d", i);
        }
    }
    return 0;
}
```

```

4.  _Bool prekid = 0;
    i = poc;
    while (i <= m && !prekid) {
        if (i > k) {
            x = x * 1.1f;
            prekid = 1;
        } else {
            if (i < t) {
                y = x + y * 0.9f;
            }
            z = y + z / 1.2f;
            i = i + n;
        }
    }
}

```

[illegible]

6. 3 0 0 5 5 4 0 0 0 0 0