

JMBAG	
Ime i prezime	

## Programiranje i programsko inženjerstvo

### Međuispit

21. studenog 2013.

**Nije dopušteno korištenje naredbe *goto*.**

**Odgovore na 1, 2. i 3. pitanje napisati na vlastitim papirima i rješenja predati u košuljici.**

#### 1. (7 bodova)

Napisati program koji s tipkovnice učitava dva prirodna broja koji predstavljaju granice zatvorenog intervala (nije potrebno provjeravati jesu li učitani brojevi prirodni niti je li prvi broj manji od drugog broja). Program treba na zaslon ispisati sve proste brojeve u zadanom intervalu, koristeći ispis kao u primjerima.

Na primjer, za učitane brojeve 101 i 110, na zaslon se ispisuje:

Prosti brojevi u intervalu [101, 110]:

Broj: 101

Broj: 103

Broj: 107

Broj: 109

Ukupno prostih brojeva: 4

, a za učitane brojeve 14 i 16, na zaslon se ispisuje poruka:

U intervalu [14, 16] nema prostih brojeva.

#### 2. (6 bodova)

S tipkovnice učitati niz znakova *niz1* ne dulji od 10 znakova, jedan znak *zn* i jedan nenegativni cijeli broj *indeks* (nije potrebno provjeravati učitane vrijednosti).

Kreirati novi niz *niz2* kopiranjem znakova iz niza *niz1*. Nakon toga, ako je *indeks* manji od duljine niza *niz2*, u nizu *niz2* promijeniti postojeći znak na poziciji *indeks* prema sljedećim pravilima:

- ako je postojeći znak slovo, nova vrijednost je učitani znak *zn*
- ako postojeći znak nije slovo niti znamenka, nova vrijednost je znak ?

Na zaslon ispisati *niz1* i *niz2*.

Na primjer, za uneseni niz A!b2C, znak 7 i poziciju 0, program treba ispisati:

Originalni niz: A!b2C, novi niz: 7!b2C

, za uneseni niz A!b2C, znak A i poziciju 1, program treba ispisati:

Originalni niz: A!b2C, novi niz: A?b2C

, a za uneseni niz A!b2C, znak A i poziciju 3, *niz2* ostaje nepromijenjen, pa program treba ispisati:

Originalni niz: A!b2C, novi niz: A!b2C

#### 3. (6 bodova)

Napisati program koji s tipkovnice učitava dimenziju (ne veću od 10) kvadratnog dvodimenzijskog polja (matrice) realnih brojeva, sve dok se ne učitava ispravna vrijednost. Zatim učitati vrijednosti elemenata prvog retka polja. Program treba preostale retke polja popuniti vrijednostima koje odgovaraju cikličkom pomaku ulijevo za 1 vrijednosti elemenata iz prethodnog retka. Nakon toga ispisati vrijednosti elemenata polja prema formatu iz prikazanog primjera.

Na primjer, za dimenziju 5 i učitane vrijednosti 3.2 6.4 8.11 9.02 1.45 program ispisuje polje na sljedeći način:

1234567890123456789012345678901234567890

	0	1	2	3	4
0	3.20	6.40	8.11	9.02	1.45
1	6.40	8.11	9.02	1.45	3.20
2	8.11	9.02	1.45	3.20	6.40
3	9.02	1.45	3.20	6.40	8.11
4	1.45	3.20	6.40	8.11	9.02

**Odgovore na 4, 5. i 6. pitanje napisati na za to predviđena mjesta uz zadatke.**

**4. (2 boda)**

Dopuniti programski odsječak tako da nakon njegovog obavljanja u polju *frekvenc* bude pohranjen broj pojavljivanja svake ocjene za jednog studenta. Ocjene su učitane kao niz znakova. Na primjer, ako je u varijablu *ocjene* učitani niz 2334555225, nakon obavljanja programskog odsječka u polju *frekvenc* moraju se nalaziti vrijednosti: 0, 3, 2, 1, 4, jer se ocjena 1 nije pojavila, ocjena 2 se pojavila 3 puta, ocjena 3 se pojavila dva puta, itd. Nije dozvoljeno korištenje pomoćnih varijabli.

```
...
int i=0, frekvenc[5]={0};
char ocjene[20];
... /* učitavanje niza znakova ocjene */

while( ) {

}

...
```

**5. (2 boda)**

Napisati naredbu kojom se, uz korištenje ternarnog operatora, može zamijeniti odsječak programa unutar pravokutnika, zadržavajuću istu funkcionalnost:

```
...
char op;
int z,g;
... /* učitavanje brojeva z i g */

if (op=='+')
    g = z = z + 1;
else {
    g = z;
    z = z - 1;
}

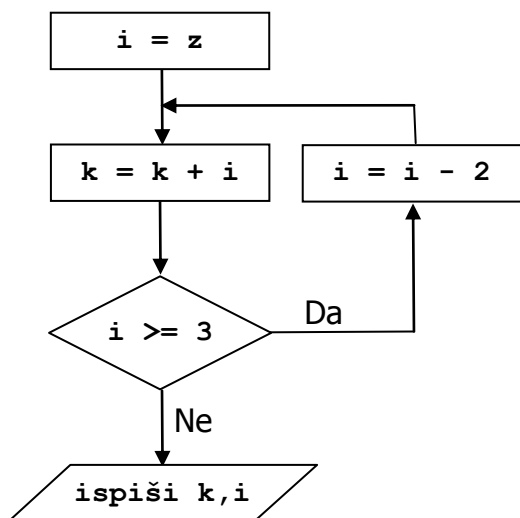
... /* ispis brojeva z i g */
```

Odgovor:

\_\_\_\_\_;

**6. (2 boda)**

U uokvirenom prostoru napisati odsječak programa koji, korištenjem **petlje s ispitivanjem uvjeta na kraju**, obavlja potpuno istu funkciju kao odsječak programa prikazan blok dijagramom. Nije dozvoljeno korištenje pomoćnih varijabli.



Rješenja:

1.

```
#include <stdio.h>
#include <math.h>
int main(void){
    int dg, gg, i, broj, brojac=0, jeProst;
    printf("Unesite donju i gornju granicu intervala brojeva: ");
    scanf("%d %d", &dg, &gg);

    for (broj=dg; broj<=gg; ++broj){
        jeProst=1;
        /* ako je broj veći od 2 i paran je, onda nije prost */
        if (broj%2 == 0 && broj > 2)
            jeProst = 0;
        /* inače ispitivati djeljivost samo s neparnim brojevima većim od 2 */
        else
            for (i=3; i<= (int) sqrt(broj); i = i+2) { /* ili pow (broj, 1./2)) */
                if (broj%i == 0) {
                    jeProst = 0;
                    break;
                }
            }
        if (jeProst) {
            brojac++;
            if (brojac==1)
                printf("Prosti brojevi u intervalu [%d, %d]:\n", dg,gg);
            printf("Broj: %d\n", broj);
        }
    }
    if (brojac>0)
        printf("Ukupno prostih brojeva: %d", brojac);
    else
        printf("\nU intervalu [%d, %d] nema prostih brojeva.", dg, gg);
    return 0;
}
```

2.

```
#include <stdio.h>
int main (void) {
    char niz1[10+1], niz2[10+1], zn;
    int dulj=0, indeks;
    printf ("Unesite niz: ");
    gets(niz1);

    printf("\nUnesite znak i indeks: ");
    scanf ("%c %d", &zn, &indeks);
    while (niz1[dulj] != '\0') {
        niz2[dulj]=niz1[dulj];
        dulj++;
    }
    niz2[dulj] = '\0';

    if (indeks<dulj) {
        if ((niz1[indeks]>='a' && niz1[indeks]<='z') ||
            (niz1[indeks]>='A' && niz1[indeks]<='Z'))
            niz2[indeks] = zn;
        else if (!(niz1[indeks]>='0' && niz1[indeks]<='9'))
            niz2[indeks] = '?';
    }
    printf("\nOriginalni niz: %s, novi niz: %s", niz1, niz2);
    return 0;
}
```

3.

```
#include <stdio.h>
#define MAXDIM 10
int main (void){
    float mat[MAXDIM][MAXDIM];
    int ns, i, j;

    do {
        printf ("Unesite broj stupaca: ");
        scanf ("%d", &ns);
    } while (ns<1 || ns> MAXDIM);

    printf ("Unesite elemente prvog retka: ");
    for (i=0; i<ns; i++)
        scanf ("%f", &mat[0][i]);

    for (i=1; i<ns; ++i){
        for (j=0; j<ns-1; ++j) {
            mat[i][j]= mat[i-1][j+1];
        }
        mat[i][ns-1]= mat[i-1][0];
    }
    /* ispis */
    printf("      ");
    for (i=0; i<ns; ++i) printf ("%6d", i);
    printf("\n");
    for (i=0; i<ns; ++i){
        printf ("%6d", i);
        for (j=0; j<ns; ++j)
            printf ("%6.2f", mat[i][j]);
        printf("\n");
    }
    return 0;
}
```

```
4. while (ocjene[i]!='\0'){
    frekvenc[ocjene[i]-'0' - 1]++;
    i++;
}
```

```
5. g = op == '+' ? ++z : z--;
```

6. Neka od rješenja:

```
i = z;
k = k + i;
if (i >= 3) {
    do {
        i = i - 2;
        k = k + i;
    } while (i >= 3);
}
printf ("%d %d\n", k, i);
```

```
i = z;
k = k + i;
do {
    if (i >= 3) {
        i = i - 2;
        k = k + i;
    }
} while (i >= 3);
printf ("%d %d\n", k, i);
```

```
i = z;
do {
    k = k + i;
    if (i < 3) break;
    i = i - 2;
} while (1);
printf ("%d %d\n", k, i);
```