

JMBAG	
Ime i prezime	

Programiranje i programsko inženjerstvo Međuispit

21. studenog 2011.

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

1. (6 bodova)

Napisati program koji učitava granice dva zatvorena intervala $[dg1, gg1]$ i $[dg2, gg2]$ cijelih brojeva. Pretpostaviti da su granice ispravno unesene. Nakon toga učitavati cijele brojeve za koje je potrebno utvrditi nalaze li se u zadanim intervalima. Učitavanje brojeva prekida se nakon učitavanja nule. Program treba ispisati koliko učitanih brojeva (ne uključujući nulu) je bilo iz intervala $[dg1, gg1]$, koliko iz intervala $[dg2, gg2]$ i koliko je brojeva iz presjeka zadanih intervala, ako presjek zadanih intervala postoji.

Primjer: za intervale $[-5, 10]$ i $[7, 20]$ i učitane brojeve: -1, 5, 3, 66, 13, 19, 8, 10, -3, 0 program ispisuje sljedeće poruke: "Interval $[-5, 10]$, učitano brojeva: 6", "Interval $[7, 20]$, učitano brojeva: 4", i kako u ovom slučaju postoji presjek intervala ispisati još i poruku "Interval presjeka $[7, 10]$, učitano brojeva: 2".

2. (6 bodova)

Napisati program koji učitava dva znakovna niza (ne dulja od 100 znakova) koristeći funkciju `gets`. Program treba ispisati poziciju i slova ako se na istoj poziciji u oba niza pojavljuje isto slovo, bez obzira radi li se o velikom ili malom slovu.

Primjer: za učitane nizove "D+2a!dfgd" i "d+2AiakKdope" program ispisuje
pozicija 1 - slova: D, d
pozicija 4 - slova: a, A
pozicija 9 - slova: d, d

3. (6 bodova)

Napisati program koji učitava red realne kvadratne matrice (maksimalni red matrice je 15). Pri definiranju maksimalnog reda matrice obavezno koristiti simboličke konstante. Red matrice učitavati sve dok se ne učitava valjana vrijednost. Učitati elemente matrice. Program treba na nulu postaviti sve elemente matrice čija je vrijednost veća od najveće vrijednosti elemenata iznad sporedne dijagonale.

Dobivenu matricu ispisati na zaslon u sljedećem obliku:

```

_ _ _ xxxx.xx _ _ _ xxxx.xx _ _ _ xxxx.xx _ _ _ xxxx.xx
_ _ _ xxxx.xx _ _ _ xxxx.xx _ _ _ xxxx.xx _ _ _ xxxx.xx
...

```

Napomena: znak `_` označava prazninu.

Primjer: za učitano matricu

4.4	2.85	0.0	3.33
6.2	1.0	6.8	4.1
2.02	8.0	3.2	-3.1
-0.5	1.0	6.8	8.1

program ispisuje:

```

4.40      2.85      0.00      3.33
6.20      1.00      0.00      4.10
2.02      0.00      3.20     -3.10
-0.50     1.00      0.00      0.00

```

Napomena: deblje su otisnute promijenjene vrijednosti.

Okrenuti stranicu →

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

4. (3 boda)

U praznom prostoru označenom pravokutnikom napisati odsječak programa tako da obavlja potpuno istu funkciju kao odsječak programa unutar pravokutnika, ali bez korištenja naredbe **goto**.

```
int main(){
    int broj, brojac=0, suma=0;
    lab1:
        scanf("%d", &broj);
        if (broj%2) {
            if ((suma + broj)>1000000) goto lab2;
            suma+=broj;
            brojac++;
        }
        if (brojac<100) goto lab1;
    lab2:
        printf("%d %d", brojac, suma);
    return 0;
}
```

5. (2 boda)

U praznom prostoru označenom pravokutnikom napisati odsječak programa tako da obavlja potpuno istu funkciju kao prikazani odsječak, ali korištenjem naredbe **switch**.

```
if (znak != 'a' && znak != 'e' && znak != 'i' && znak != 'o' && znak != 'u')
    printf("Uneseni znak %c nije samoglasnik", znak);
else
    printf("Uneseni znak %c je samoglasnik", znak);
```

6. (2 boda)

U prostor nad crtom u danom programskom odsječku napisati izraz kojim se izračunava vrijednost 27. bita učitano broja (vrijednost treba biti 0 ili 1). Primjer broja kod kojeg je vrijednost 27. bita jednaka 1:

bit	32	31	...	28	27	26	...	2	1
vrijednost	0	1	...	0	1	1	...	0	0

```
int broj, vrijednost;
printf("Unesite broj: ");
scanf("%d", &broj);
```

```
vrijednost = _____;
printf("Vrijednost 27. bita = %d", vrijednost);
```

Rješenja:

1. 6 bodova

```
int main() {
    int dg1, gg1, dg2, gg2, presjekdg, presjekgg, imapresjek=0, broj;
    int brojac1=0, brojac2=0, brojacp=0;
    printf("Unesite granice intervala: ");
    scanf("%d %d %d %d", &dg1, &gg1, &dg2, &gg2);

    presjekdg=dg1<dg2?dg2:dg1;
    presjekgg=gg1<gg2?gg1:gg2;
    imapresjek=presjekdg>presjekgg?0:1;
    printf("Ucitavajte brojeve ...\n");
    do {
        scanf("%d", &broj);
        if (broj==0) break;
        if (broj>=dg1 && broj<=gg1) brojac1++;
        if (broj>=dg2 && broj<=gg2) brojac2++;
        if (imapresjek && broj>=presjekdg && broj<=presjekgg) brojacp++;
    } while(1);
    printf("\nInterval [%d, %d], ucitano brojeva: %d", dg1, gg1, brojac1);
    printf("\nInterval [%d, %d], ucitano brojeva: %d", dg2, gg2, brojac2);
    if (imapresjek)
        printf("\nInterval presjeka [%d, %d], ucitano brojeva: %d", presjekdg, presjekgg, brojacp);
    return 0;
}
```

Alternativno rješenje koje ne ispisuje ništa ako presjek intervala postoji ali NEMA zajedničkih elemenata.

```
int main() {
    int dg1, gg1, dg2, gg2, presjekdg, presjekgg, broj;
    int brojac1=0, brojac2=0, brojacp=0;
    printf("Unesite granice intervala");
    scanf("%d %d %d %d", &dg1, &gg1, &dg2, &gg2);
    printf("Ucitavajte brojeve ...\n");
    do {
        scanf("%d", &broj);
        if (broj==0) break;
        if (broj>=dg1 && broj<=gg1) brojac1++;
        if (broj>=dg2 && broj<=gg2) brojac2++;
        if (broj>=dg1 && broj<=gg1 && broj>=dg2 && broj<=gg2) brojacp++;
    } while(1);

    printf("\nInterval [%d, %d], ucitano brojeva: %d", dg1, gg1, brojac1);
    printf("\nInterval [%d, %d], ucitano brojeva: %d", dg2, gg2, brojac2);
    if (brojacp){
        presjekdg=dg1<dg2?dg2:dg1;
        presjekgg=gg1<gg2?gg1:gg2;
        printf("\nInterval [%d, %d], ucitano brojeva: %d", presjekdg, presjekgg, brojacp);
    }
    return 0;
}
```

2. 6 bodova

```
#define MAX_NIZ 100
int main() {
    char a[MAX_NIZ + 1], b[MAX_NIZ + 1];
    int i=0;
    gets(a);
    gets(b);
    while (a[i] != '\0' && b[i] != '\0') {
        if ((a[i] >= 'A' && a[i] <= 'Z' && (a[i] == b[i] || a[i]+32 == b[i])) ||
            (a[i] >= 'a' && a[i] <= 'z' && (a[i] == b[i] || a[i]-32 == b[i])))
            printf("pozicija %d - slova: %c, %c\n", i+1, a[i], b[i]);
        i++;
    }
    return 0;
}
```

3. 6 bodova

```
#define MAXR 15
int main() {
    int n, i, j;
    float A[MAXR][MAXR], najveci;
    do {
        printf("Rang matrice A: ");
        scanf("%d", &n);
    } while(n<1 || n> MAXR);

    printf("\nUcitajte elemente matrice A\n");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++) {
            scanf("%f", &A[i][j]);
            if (i==0 && j==0) najveci = A[i][j];
            else if ((i+j)<(n-1) && A[i][j] > najveci) najveci = A[i][j];
        }

    for(i=0;i<n;i++)
        for(j=n-i-1;j<n;j++)
            if (A[i][j] > najveci) A[i][j] = 0.f;

    printf ("\n");
    for(i=0;i<n;i++) {
        for(j=0;j<n;j++)
            printf (" %7.2f", A[i][j]);
        printf ("\n");
    }
    return 0;
}
```

4. 3 boda

```
int main(){
    int broj, brojac=0,suma=0;
    do {
        scanf("%d", &broj);
        if (broj%2) {
            if ((suma + broj)>1000000) break;
            suma+=broj;
            brojac++;
        }
    } while (brojac<100);
    printf("%d %d", brojac, suma);
    return 0;
}
```

5. 2 boda

```
switch(znak){
    case 'a':
    case 'e':
    case 'i':
    case 'o':
    case 'u':
        printf("Uneseni znak %c je samoglasnik", znak);
        break;
    default:
        printf("Uneseni znak %c nije samoglasnik", znak);
}

Ili
switch(znak != 'a' && znak != 'e' && znak != 'i' && znak != 'o' && znak != 'u'){
    case 0:
        printf("Uneseni znak %c je samoglasnik", znak);
        break;
    case 1:
        /*mogao je pisati i default*/
        printf("Uneseni znak %c nije samoglasnik", znak);
}
}
```

6. 2 boda

```
vrijednost = (broj>>(27-1)) & 1;
ili
vrijednost = broj & (int)pow(2., 27-1) ? 1 : 0;
ili
vrijednost = (broj>>(27-1)) % 2;
```