



Универзитет „Св. Кирил и Методиј“ - Скопје
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

Аудиториски вежби 1

Структури

Објектно ориентирано програмирање

Задача 1

Да се напише програма која ќе споредува два датуми (ден, месец, година) и ќе ја пресмета разликата во денови од едниот до другиот датум. Пресметките да се реализираат во посебни функции.

За датумот да се дефинира посебна структура datum.

Задача 1

Решение 1/3

```
#include <stdio.h>

struct datum {
    int den;
    int mesec;
    int godina;
};

typedef struct datum datum;
```

Задача 1

Решение 2/3

```
short sporedba(datum d1, datum d2) {
    if (d1.den == d2.den && d1.mesec == d2.mesec && d1.godina == d2.godina)
        return 0;
    if (d1.godina > d2.godina)
        return 1;
    else if (d1.godina < d2.godina)
        return -1;
    else {
        if (d1.mesec > d2.mesec)
            return 1;
        else if (d1.mesec < d2.mesec)
            return -1;
        else {
            if (d1.den > d2.den)
                return 1;
            else
                return -1;
        }
    }
}
```

Задача 1

Решение 3/3

```
long razlika(datum d1, datum d2) {
    long denovi;
    denovi = d1.den - d2.den;
    denovi += (d1.mesec - d2.mesec) * 30;
    denovi += (d1.godina - d2.godina) * 360;
    return denovi;
}

int main() {
    datum d1 = { 14, 12, 1989 };
    datum d2;
    d2.den = 16;
    d2.mesec = 12;
    d2.godina = 1989;
    if (sporedba(d1, d2) == 0)
        printf("Datumite se isti.\n");
    else if (sporedba(d1, d2) == 1)
        printf("Razlikata pomegu datumite e %d dena.\n", razlika(d1, d2));
    else
        printf("Razlikata pomegu datumite e %d dena.\n", razlika(d2, d1));
    return 0;
}
```

Задача 2

Да се напише програма која ќе го пресметува векторскиот и скаларниот производ на два вектори. Векторите се претставени со координати во тродимензионален координатен систем. Скаларниот и векторскиот производ да се пресметуваат со посебни функции. *За вектор да се дефинира посебна структура вектор.*

Задача 2

Решение 1/2

```
#include <stdio.h>

struct vektor {
    float x;
    float y;
    float z;
};

typedef struct vektor vektor;

float skalaren_proizvod(vektor v1, vektor v2) {
    return v1.x * v2.x + v1.y * v2.y + v1.z * v2.z;
}

vektor vektorski_proizvod(vektor v1, vektor v2) {
    vektor v;
    v.x = v1.y * v2.z - v1.z * v2.y;
    v.y = v1.z * v2.x - v1.x * v2.z;
    v.z = v1.x * v2.y - v1.y * v2.x;
    return v;
}
```

Задача 2

Решение 2/2

```
int main() {  
    vektor v1 = { 2, 4, 6 };  
    vektor v2 = { 3, 5, 9 };  
    vektor v;  
    v = vektorski_proizvod(v1, v2);  
    printf("v1 * v2 = %.2f\n", skalaren_proizvod(v1, v2));  
    printf("v1 x v2 = [%.2f, %.2f, %.2f]\n", v.x, v.y, v.z);  
    return 0;  
}
```

Задача 3

Да се напише програма во која ќе се дефинира структура за претставување комплексни броеви. Потоа да се напишат функции за собирање, одземање и множење на два комплексни броеви. Програмата да се тестира во главна програма во која се вчитуваат два комплексни броја од стандарден влез.

Задача 3

Решение 1/3

```
#include <stdio.h>

typedef struct kompleksen {
    float realen;
    float imag;
} kompl;

kompl soberi(kompl a, kompl b) {
    kompl c = a;
    c.realen += b.realen;
    c.imag += b.imag;
    return c;
}

kompl odzemi(kompl *pok1, kompl *pok2) {
    kompl c = *pok1;
    c.realen -= (*pok2).realen;
    c.imag -= (*pok2).imag;
    return c;
}
```

Задача 3

Решение 2/3

```
void mnozi(kompl a, kompl b, kompl *c) {
    c->realen = a.realen * b.realen - a.imag * b.imag;
    c->imag = a.realen * b.imag + a.imag * b.realen;
}

void pecati(kompl *pok) {
    printf("%.2f", pok->realen);
    if (pok->imag >= 0)
        printf("+j%.2f\n", pok->imag);
    else
        printf("-j%.2f\n", abs(pok->imag));
}
```

Задача 3

Решение 3/3

```
int main() {
    kompl a, b, c;
    printf("Vnesete dva kompleksni broevi:\n");
    printf("Prv broj:\n");
    printf("Realen: ");
    scanf("%f", &a.realen);
    printf("Imaginaren: ");
    scanf("%f", &a.imag);
    printf("Prv broj:\n");
    printf("Realen: ");
    scanf("%f", &b.realen);
    printf("Imaginaren: ");
    scanf("%f", &b.imag);
    printf("Gi vnesovte broevite:\n");
    pecati(&a);
    pecati(&b);
    printf("Nivniot zbir e:\n");
    c = soberi(a, b);
    pecati(&c);
    printf("Nivnata razlika e:\n");
    c = odzemi(&a, &b);
    pecati(&c);
    printf("Nivniot proizvod e:\n");
    mnozi(a, b, &c);
    pecati(&c);
    return 0;
}
```

Задача 4

Од стандарден влез се читаат непознат број податоци за студенти. Податоците се внесуваат така што во секој ред се дава:

- името
- презимето
- бројот на индекс (формат ххyzzzz)
- четири броја (поени од секоја задача)

со произволен број празни места или табулатори меѓу нив. Да се напише програма која ќе испечати список на студенти, каде во секој ред ќе има: презиме, име, број на индекс, вкупен број на бодови сортиран според бројот на бодови. При тоа имињата и презимињата да се напишат со голема почетна буква.

Задача 4

Решение 1/3

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

struct kandidat {
    char ime[15];
    char prezime[20];
    int index;
    int bodovi;
};

void norm(char *s) {
    //Prva буква golema, ostanatite mali
    *s = toupper(*s);
    while (*(++s) != '\0')
        *s = tolower(*s);
}
```

Задача 4

Решение 2/3

```
void sort(struct kandidat a[], int n) {
    int i, j;
    struct kandidat s;
    for (i = 0; i < n; i++)
        for (j = 0; j < n - i - 1; j++)
            if (a[j].bodovi < a[j + 1].bodovi) {
                s = a[j];
                a[j] = a[j + 1];
                a[j + 1] = s;
            }
}
```

Задача 4

Решение 3/3

```
int main() {
    struct kandidat st[50];
    int i, n;
    printf("Vnesete broj na studenti: ");
    scanf("%d", &n);
    for (i = 0; i < n; ++i) {
        printf("Nov student:\n");
        printf("Ime: ");
        scanf("%s", &st[i].ime);
        printf("Prezime: ");
        scanf("%s", &st[i].prezime);
        printf("Indeks: ");
        scanf("%d", &st[i].index);
        printf("Poeni (z1 z2 z3 z4): ");
        int j, zadaca;
        st[i].bodovi = 0;
        for(j = 0; j < 4; j++) {
            scanf("%d", &zadaca);
            st[i].bodovi += zadaca;
        }
        norm(st[i].ime);
        norm(st[i].prezime);
    }
    sort(st, n);
    for (i = 0; i < n; i++) {
        printf("%d. %s %s\t%d\t%d\n", i + 1, st[i].ime, st[i].prezime, st[i].
            index, st[i].bodovi);
    }
    return 0;
}
```


Задача 5

Да се напише програма која од стандарден влез ќе чита податоци за држави и на екран ќе го отпечати името и презимето на претседателот на државата чиј што главен град има најмногу жители.

- Податоци за држава: име, претседател, главен град и број на жители
- Податоци за град: име и број на жители
- Податоци за претседател: име, презиме и политичка партија

Задача 5

Решение 1/3

```
#include <stdio.h>

typedef struct grad {
    char ime[30];
    long brZiteli;
} grad;

typedef struct pretsedatel {
    char ime[20];
    char prezime[20];
    char partija[20];
} pret;

typedef struct drzava {
    char ime[30];
    pret pretsedatel;
    long brZiteli;
    grad glaven;
} drzava;
```

Задача 5

Решение 2/3

```
int main() {
    drzava d[20];
    int n, i, maxi, max;
    printf("Vnesete broj na drzavi:\n");
    scanf("%d", &n);
    for (i = 0; i < n; ++i) {
        printf("Drzava: ");
        scanf("%s", &d[i].ime);
        printf("Pretsedatel:\n");
        printf("Ime: ");
        scanf("%s", &d[i].pretsedatel.ime);
        printf("Prezime: ");
        scanf("%s", &d[i].pretsedatel.prezime);
        printf("Partija: ");
        scanf("%s", &d[i].pretsedatel.partija);
        printf("Broj na ziteli: ");
        scanf("%d", &d[i].brZiteli);
        printf("Glaven grad: ");
        scanf("%s", &d[i].glaven.ime);
        printf("Ziteli vo gradot: ");
        scanf("%d", &d[i].glaven.brZiteli);
    }
}
```

Задача 5

Решение 3/3

```
maxi = 0;
max = d[maxi].glaven.brZiteli;
for (i = 0; i < n; ++i)
    if (d[i].glaven.brZiteli > max) {
        max = d[i].glaven.brZiteli;
        maxi = i;
    }
printf(
    "Imeto na pretsedatelot na drzavata so najgolem glaven grad e: %s %s
    \n",
    d[maxi].pretsedatel.ime, d[maxi].pretsedatel.prezime);
return 0;
}
```

Задача 4 да се модифицира така што во структурата со која е опишан студентот да се додаде низа од предмети (не повеќе од 10) со оценки од испит. Да се напише програма која ќе отпечати список со студенти сортирани според нивниот просек, почнувајќи од студентот со најголем просек. Податоците за предметот се: име на предметот и оценка по предметот.

Материјали и прашања

Предавања, аудиториски вежби, соопштенија
courses.finki.ukim.mk

Изворен код на сите примери и задачи
<https://github.com/tdelev/OOP/tree/master/latex/src>

Прашања и дискусија
forum.finki.ukim.mk