Lesson 1
Structs

Object oriented programming

Write a programm that will compare two dates (day, month, year) and will compute the difference in days between them. Implement the computations in separate functions.
Define a struct for date date.

```
#include <stdio.h>

struct date {
    int day;
    int month;
    int year;
};

typedef struct date date;
```

```
short compare(date d1, date d2) {
    if (d1.day == d2.day && d1.month == d2.month && d1.year == d2.year)
        return 0;
    if (d1.year > d2.year)
        return 1;
    else if (d1.year < d2.year)
        return -1;
    else {
        if (d1.month > d2.month)
            return 1;
        else if (d1.month < d2.month)
            return -1;
        else {
            if (d1.day > d2.day)
                return 1;
            else
                return -1;
        }
    }
}
```

Problem 1
Solution 3/3

```c
long difference(date d1, date d2) {
    long days;
    days = d1.day - d2.day;
    days += (d1.month - d2.month) * 30;
    days += (d1.year - d2.year) * 360;
    return days;
}

int main() {
    date d1 = { 14, 12, 1989 };
    date d2;
    d2.day = 16;
    d2.month = 12;
    d2.year = 1989;
    if (compare(d1, d2) == 0)
        printf("Dates are equal.\n");
    else if (compare(d1, d2) == 1)
        printf("Difference between dates is %d days.\n", difference(d1, d2));
    else
        printf("Difference between dates is %d days.\n", difference(d2, d1));
    return 0;
}
```

Write a program that will comptute the vector and the scalar product of two vectors. Vectors are represented with coordinates in threedimensional Cartesian coordinate system. Compute the products in separate functions.

Define a struct for vector named `vector`.

```c
#include <stdio.h>

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

typedef struct vector vector;

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

vector vector_product(vector v1, vector v2) {
    vector 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;
}
```

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

Write a struct for representing complex numbers. Then implement functions for addition, substruction and multiplication of two complex numbers. Test the functions in a main program where you read two complex numbers from standard input.

Problem 3
Solution 1/3

```c
#include <stdio.h>

typedef struct complex_number {
    float real;
    float imag;
} comp;

comp add(comp a, comp b) {
    comp c = a;
    c.real += b.real;
    c.imag += b.imag;
    return c;
}

comp substract(comp *pok1, comp *pok2) {
    comp c = *pok1;
    c.real -= (*pok2).real;
    c.imag -= (*pok2).imag;
    return c;
}
```

```c
void multiply(comp a, comp b, comp *c) {
    c->real = a.real * b.real - a.imag * b.imag;
    c->imag = a.real * b.imag + a.imag * b.real;
}

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

```c
int main() {
    comp a, b, c;
    scanf("%f %f", &a.real, &a.imag);
    scanf("%f %f", &b.real, &b.imag);
    print(&a);
    print(&b);
    printf("a + b\n");
    c = add(a, b);
    print(&c);
    printf("a - b\n");
    c = substract(&a, &b);
    print(&c);
    printf("a * b\n");
    multiply(a, b, &c);
    print(&c);
    return 0;
}
```

Problem 4

Read from standard input data for unknown number of students.
Each row of the data is in following format:

- first name
- last name
- number (format xxyzzzz)
- four numbers (points for each problem)

separated with tab space.
Write a program that will print list of students, where each row will
have: last name, first name, number, and total points sorted by the
number of points. BTW the names should be printed with first
capital letter.

Problem 4
Solution 1/3

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

struct student {
    char first_name[15];
    char last__name[20];
    int number;
    int points;
};

void norm(char *s) {
    // First letter uppercase, others lowercase
    *s = toupper(*s);
    while (*(++s) != '\0')
        *s = tolower(*s);
}
```

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

```
int main() {
    struct student st[50];
    int i, n;
    scanf("%d", &n);
    for (i = 0; i < n; ++i) {
        scanf("%s", &st[i].first_name);
        scanf("%s", &st[i].last__name);
        scanf("%d", &st[i].number);
        int j, zadaca;
        st[i].points = 0;
        for(j = 0; j < 4; j++) {
            scanf("%d", &zadaca);
            st[i].points += zadaca;
        }
        norm(st[i].first_name);
        norm(st[i].last__name);
    }
    sort(st, n);
    for (i = 0; i < n; i++) {
        printf("%d. %s %s\t%d\t%d\n", i + 1, st[i].first_name, st[i].last__name,
            st[i].number, st[i].points);
    }
    return 0;
}
```

Write a program that will read from standard input data for
countries and will print on the standard output the name of the
president of the country whose capital has largest population.

- Data for country: name, president, capital and population
- Data for city: name and population
- Data for president: name, political party

```c
#include<stdio.h>

typedef struct city {
    char name[30];
    long population;
} city;

typedef struct president {
    char name[20];
    char party[20];
} pres;

typedef struct country {
    char name[30];
    pres president;
    long population;
    city capital;
} country;
```

```c
int main() {
    country d[20];
    int n, i, maxi, max;
    scanf("%d", &n);
    for (i = 0; i < n; ++i) {
        scanf("%s", &d[i].name);
        printf("president:\n");
        scanf("%s", &d[i].president.name);
        scanf("%s", &d[i].president.party);
        scanf("%d", &d[i].population);
        scanf("%s", &d[i].capital.name);
        scanf("%d", &d[i].capital.population);
    }
    maxi = 0;
    max = d[maxi].capital.population;
    for (i = 0; i < n; ++i)
        if (d[i].capital.population > max) {
            max = d[i].capital.population;
            maxi = i;
        }
    printf(
            "Name of the president of the country with the largest capital is: %
                s\n",
            d[maxi].president.name);
    return 0;
}
```

Modify problem 4 so the struct for student has an array of subjects (not more than 10) with grades from exams. Write a program that will print list of students sorted by their avarage grade in descending order. Data for the subject is: name of the subject and grade.

Materials and Questions

Lectures, exsercises and announcements
courses.finki.ukim.mk

Source code of all examples and problems
https://github.com/tdelev/SP/tree/master/latex/src

Questions and discussion
forum.finki.ukim.mk