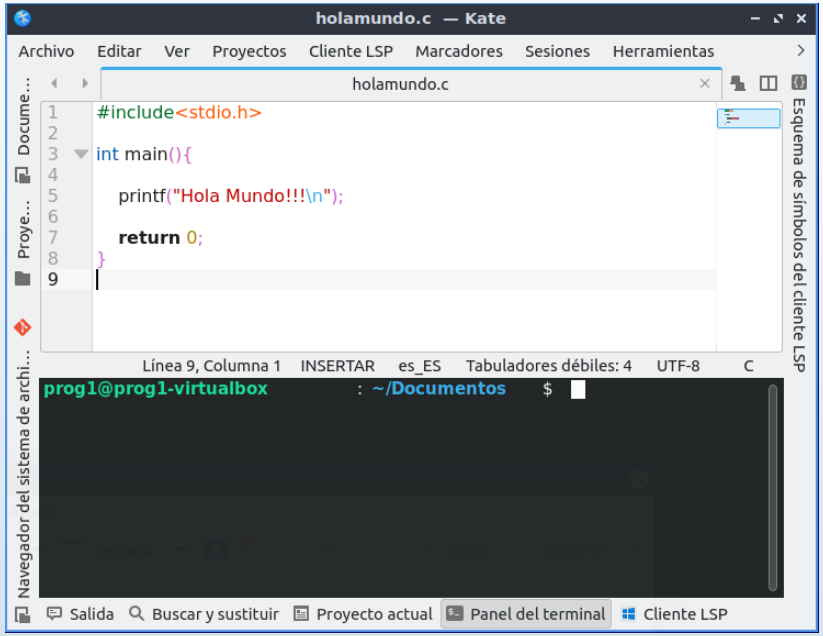

Programming 1

Lesson 7. Structured data types: structures - Exercises

Degree in Computer Engineering

Exercise 1

You want to store the information related to the subjects of a degree. The information for each subject is: code, name, semester, type (basic, compulsory, optional), course, code of the coordinating teacher and codes of the subjects you need to have passed before this subject (5 as maximum). You also want to store the teacher's information: their code, their name, and the codes of their coordinated subjects (3 subjects as maximum)

- Design the necessary data structures to store all the information having into account that the degree is formed by 45 subjects and there are 60 teachers in total
- Design a module that displays on screen a list of compulsory subjects in the first semester(only name and course)
- Design a module that requests a teacher code and displays on the screen the name of all their coordinated subjects.

Exercise 1 - Solution


```
#define NSubjectCodes 5
#define NSubjectCoord 3
#define NSubjects 45
#define NTeachers 60

enum T_Subject_Type {basic, compulsory, optional};

typedef struct {
    int code;
    char name [20];
    int semester;
    enum T_Subject_Type type;
    int course;
    int coordinator;
    int subjectCodes [NSubjectCodes];
} TSubject;

typedef struct {
    int code;
    char name [50];
    int coordSubjects[NSubjectCoord];
} TTeacher;

typedef TTeacher TTeachers [NTeachers];
typedef TSubject TSubjects [NSubjects];
```

Exercise 1 - Solution


```
void listCompulsory (TSubjects subjects, int nsubj) {
    int i;
    for (i=0; i<nsubj; i++) {
        if (subjects[i].type == compulsory && subjects[i].semester == 1)
             printf("Subject: %s\tCourse: %d\n", subjects[i].name, subjects[i].course);
    }
}
```

```
void listSubjects(TSubjects subjects, int nsubj, TTeachers teachers, int nteach, int code) {
    int i, j, k;
    bool found = false;
    bool foundsub;
    //Check if teacher exists
    while (i < nteach && !found) {
        if (teachers[i].code == code) {
            found = true;
            j = 0;
            while (j < NSubjectCoord && teachers[i].coordSubjects[j] > 0) {
                foundsub = false;
                k = 0;
                while (k < nsubj && !foundsub) {
                    if (subjects[k].code == teachers[i].coordSubjects[j]) {
                        foundsub = true;
                        printf("%s\n", subjects[k].name);
                    }
                    k++;
                }
                j++;
            }
        }
        i++;
    }
}
```

Exercise 2

You want to design a program to manage the medicines in a pharmacy. The information for each medicine is: code (integer), name, description, purchase price, sales price, active component, and the code of the laboratories that produce it. Each medicine can be manufactured by 1 or more laboratories (for legal reasons up to a maximum of 10). The laboratories are interested in storing the following information: code, name, registered office. We have a maximum of 50 laboratories.

- Design the necessary data structures to manage all the information having into account that there are 500 medicines at most.
- Design a module that calculates and returns the code of the medicine with the major benefit to the pharmacy.
- Design a module that shows a list on screen with the name of the medicines produced by each laboratory. This module must also return the name of the laboratory that produces the most variety of medicines, the number of different medicines produced by that laboratory, the name of the laboratory producing the least number of medicines and the number of medicines it produces.

Exercise 2 - Solution


```
#define NManulabs 10
#define NLabs 50
#define NMedicines 500

typedef struct {
    int code;
    char name [20];
    char description [50];
    float purchase_price;
    float sales_price;
    char active_component [30];
    int laboratories [NManulabs];
} TMedicine;

typedef struct {
    int code;
    char name [50];
    char registeredOffice [50];
} TLab;

typedef TMedicine TMedicines [NMedicines];
typedef TLab TLabs [NLabs];
```

Exercise 2 - Solution


```
int majorBenefit (TMedicines medicines, int nmed) {
    int i;
    float benefit, maxBenefit = 0.0;
    int medicine = -1;

    for (i=0; i<nmed; i++) {
        benefit = medicines[i].sales_price - medicines[i].purchase_price;
        if (benefit > maxBenefit) {
            maxBenefit = benefit;
            medicine = i;
        }
    }
    return medicine;
}
```

Exercise 2 - Solution

```
void listMedicines(TMedicines medicines, int nmed, TLabs laboratories, int nlaboratories, int* maxLab, int* maxNumber, int* minLab, int* minNumber) {
    int imed, ilab, i;
    int maxMed = -1, minMed = NMedicines + 1;
    int maxl = -1, minl = -1, count;
    bool found;

    for (ilab = 0; ilab < nlaboratories; ilab++) {
        printf("Laboratory: %s\n", laboratoires[ilab].name);
        count = 0;
        for (imed = 0; imed < nmed; imed++) {
            found = false;
            i = 0;
            while (!found && i < NManulabs) {
                if (laboratories[ilab].code == medicines[imed].laboratories[i]) {
                    found = true;
                    count++;
                    printf("\t%s\n", medicines[imed].name);
                }
                i++;
            }
        }
        if (count > maxMed) {
            maxMed = count;
            maxl = ilab;
        }
        if (count < minMed) {
            minMed = count;
            minl = ilab;
        }
    }
    *maxLab = maxl;
    *minLab = minl;
    *maxNumber = maxMed;
    *minNumber = minMed;
}
```

Exercise 3

Implement a program to manage a private library. The information for each book is: title, ISBN (alphanumeric code), author, genre (thriller, romance, history, poetry, children) and publisher. We are also interested in storing some information about the publishers: name, address and e-mail. There are 40 publishers.

- Design the data structures to manage all the information, considering that we need to store 500 books at most.
- Design a module to calculate and return the genre with the greatest amount of books, as well as the amount of books of this genre.
- Design a module that asks the title of a book to the user, and shows on the screen the email of the publisher of this book, as well as the title of all the books in the library that are published by this publisher.

You can make use of functions strcpy() and strcmp() if needed.

Exercise 3 - Solution


```
#define NBooks 500
#define NPublishers 40

enum T_Genre_Type {thriller, romance, history, poetry, children};

typedef struct {
    char title [40];
    char author [40];
    enum T_Genre_Type genre;
    char publisher [40];
} TBook;

typedef struct {
    char name [40];
    char address [100];
    char email [20];
} TPublisher;

typedef TBook TBooks [NBooks];
typedef TPublisher TPublishers [NPublishers];
```

Exercise 3 - Solution


```
void greatest (TBooks books, int nbook, enum T_Genre_Type* genre, int* amount) {
    int i, ibook, count, maxbook = 0;
    for (i=thriller; i <= children; i++) {
        count = 0;
        for (ibook=0; ibook < nbook; ibook++) {
            if (books[ibook].genre == i)
                 count++;
        }
        if (count > maxbook) {
            maxbook = count;
            *genre = i;
        }
    }

    *amount = maxbook;
}
```

Exercise 3 - Solution


```
void showbytitle (TBooks books, int nbook, TPublishers publishers, int npub) {
    char title [40];
    int ibook, ipub, i;
    bool found = false;

    printf("Enter the title of the book: ");
    scanf(" %[^\n]s", title);

    // Search the book
    ibook = 0;
    while (!found && ibook < nbook) {
        if (!strcmp(title, books[ibook].title))
            found = true;
        else
            ibook++;
    }

    if (found) {
        // The book has been found
        printf("The publisher is %s\n", books[ibook].publisher);

        // Search for the publisher
        found = false;
        ipub = 0;
        while (!found && ipub < npub) {
            if (!strcmp(books[ibook].publisher, publishers[ipub].name)) {
                found = true;
                printf("His/Her email is %s\n", publishers[ipub].email);
            }
            else
                ipub++;
        }

        printf(“All the books published by this publisher are: \n");
        // List the books
        for (i=0; i < nbook; i++) {
            if (!strcmp(books[i].publisher, publishers[ipub].name) {
                printf("\t%s\n", books[i].title);
            }
        }
    }
    else {
        printf("This book is not in the library\n");
    }
}
```