

SOLUTIONS FOR EXEMPLARY TASKS
illustrating processing the data in the form of an **array of structures**

- TASK 1** Define a struct type and a 100-element array to store information about **books in the library** (title, author, index, price). Write a function that will display all the data of "*cheap books*" on the screen (ie books with price < 10).
- TASK 2** Define the struct type and the 50-element array of structures allowing to store information about **factory employees** (name, position, gender, salary) Write a function that calculates the average salary of women in this factory.
- TASK 3** Define the struct type that stores the **data of the computer set** in the store (set name = 50 characters, processor type = 20 characters, clock frequency = positive integer; price = real number) and a 20-element array of such structures. Write a function that determines the index of the cheapest set with the "AMD Athlon" processor.
- TASK 4** Define the type of struct type that stores the employee's **personal data** (surname - 20 characters, first name - 15 characters, salary - real number; gender - the character 'm' or the character 'f') and a 100-element array of such structures. Write a function to check, if the average salary in a given company is greater than 3000.
- TASK 5** Define the type of structure that stores the **reservation data in the hotel** (booking date = 30 characters, guest name = 25 characters; room type = letter; number of beds = an integer) and a 100-element array of such structures. Write a function that counts the number of reservations for the name "Kowalski".
- TASK 6** Define the type of structure that stores **data about bus connections** (destination_city: 30 characters; distance: unsigned integer; departure_time: real number; price - float number) and a 500-element array of such structures. Write a function that determines the average ticket price for a route shorter than 10km.
-

1. Define a struct type and a 100-element array to store information about **books in the library** (title, author, index, price). Write a function that will display all the data of "*cheap books*" on the screen (ie books with price < 10).

```
#include <stdio.h>

// definition of a structural type
struct Book
{
    char title[50];
    char author[30];
    unsigned index;
    float price;
};

// definition of a 100-element array of structures
Book library[100];

// definition of the function displaying "cheap" books
// (with the price lower then 10 PLN)
void DISPLAY_CHEAP_BOOKS( Book lib[], int book_counter)
{
    bool occurs = false ;
    printf( "\n List of cheap books: " );
    for( int i=0; i<book_counter; i++)
        if( lib[i].price<10.00 )
        {
            printf("\n");
            printf("\n Title: %s" , lib[i].title );
            printf("\n Author: %s" , lib[i].author );
            printf("\n Index: %u" , lib[i].index );
            printf("\n Price: %.2f", lib[i].price );
            occurs = true ;
        }
    if( !occurs )
        printf("\n The list is empty (no cheap books) ");

    printf("\n Press the ENTER key ");
    getchar();
}
```

2. Define the struct type and the 50-element array of structures allowing to store information about **factory employees** (name, position, gender, salary) Write a function that calculates the average salary of women in this factory.

```
#include <stdio.h>

// definition of a structural type
struct Employee
{
    char   name[30];
    char   position[30];
    char   gender;
    float  salary;
};

// definition of a 50-element array of structures
Employee list_of_factory_workers[50];

// function calculating the average salary of women
float AVERAGE_FEMALE_SALARY( Employee list[], int employee_counter )
{
    float sum=0;
    int female_counter=0;
    for( int i=0; i<employee_counter; i++ )
        if( list[i].gender=='f' )           // 'f' like female
        {
            sum += list[i].salary;
            female_counter++;
        }
    if( female_counter==0 )
        return -1.0; // Number of women=0. Can't count the average

    average float ;
    average = sum/female_counter;
    return average;
}
```

3. Define the struct type that stores the **data of the computer set** in the store (set_name = 50 characters, processor type = 20 characters, clock frequency = positive integer; price = real number) and a 200-element array of such structures. Write a function that determines the index of the cheapest set with the "AMD Athlon" processor.

```
#include <string.h>

struct ComputerSet
{
    char set_name[51];
    char processor_type[21];
    unsigned clock;
    float price;
} ;

ComputerSet store[200];

int cheap_AMD( ComputerSet store[], int array_size )
{
    int index = -1;
    float lowest_price;
    for( int i=0; i<array_size; i++ )
        if( strcmp( store[i].processor_type, "AMD Athlon" ) == 0 )
            if( index==-1 || store[i].price<lowest_price )
            {
                index = i;
                lowest_price = store[i].price;
            }

    return index;
}
```

4. Define the type of struct type that stores the employee's **personal data** (surname - 20 characters, first name - 15 characters, salary - real number; gender - the character 'm' or the character 'f') and a 100-element array of such structures. Write a function to check, if the average salary in a given company is greater than 3000.

```
struct EmployeeData
{
    char    surname [21];
    char    name [16];
    float   salary;
    char    gender;
} ;

EmployeeData list_of_employees[100];

bool isSalaryGreater( EmployeeData employees_list[],
                     int number_of_employees )
{
    if( number_of_employees<=0 )
        return false ;

    // calculating the sum of the salaries
    float sum=0;
    for ( int i=0; i<number_of_employees; i++)
        sum += employees_list[i].salary;
    // check the average salary
    if( sum/number_of_employees > 3000 )
        return true ;
    else
        return false ;
}
```

5. Define the type of structure that stores the **reservation data in the hotel** (booking_date = 30 characters, guest name = 25 characters; room_type = letter; number_of_beds = an integer) and a 100-element array of such structures. Write a function that counts the number of reservations for the name "Kowalski".

```
#include <string.h>

struct Reservation
{
    char booking_date[31];
    char guest_name[26];
    char room_type;
    int  number_of_beds;
} ;

Reservation reservation_list[100];

int CountReservationsForKowalski( Reservation reservation_list[] )
{
    int counter=0;
    for( int i=0; i<100; i++ )
        if( strcmp( reservation_list[i].guest_name, "Kowalski") == 0 )
            counter ++;
    return counter;
}
```

6. Define the type of structure that stores data about **bus connections**

(destination_city: 30 characters; distance: unsigned integer;

departure_time: real number; price - float number)

and a 500-element array of such structures.

Write a function that determines the average ticket price for a route shorter than 10km.

```
struct BusConnection
{
    char    destination_city[31];
    unsigned distance;
    float    departure_time;
    float    price;
};

BusConnection timetable[500];

float Average price ( BusConnection timetable[],
                     int number_of_connections )
{
    float sum_of_selected=0;
    int number_of_selected=0;
    for( int i=0; i<number_of_connections; i++ )
        if( timetable[i].distance <10)
        {
            sum_of_selected += timetable[i].price;
            number_of_selected++;
        }
    if (number_of_selected == 0)
        return -1;
    else
        return sum_of_selected/number_of_selected;
}
```