Introduction to Programming I

Lab 4

Alexey Shikulin, Munir Makhmutov, Sami Sellami and Furqan Haider

Structures, Union and Recursive Functions

Exercise 1

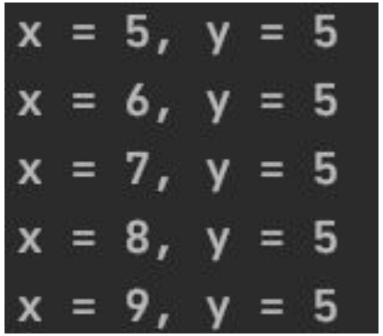
What is the expected output of this program?

```
#include <stdio.h>
void func() {
      static int x = 5;
      int y = 5;
      while (y < 10 && x < 10) {
            printf("x = %d, y = %d\n", x, y);
            x++;
            y++;
            func();
    }
}
int main() {
       func();
}
```

Exercise 1: Solution

What is the expected output of this program?

```
#include <stdio.h>
void func() {
      static int x = 5;
      int y = 5;
      while (y < 10 && x < 10) {
            printf("x = %d, y = %d\n", x, y);
            x++;
            y++;
            func();
    }
}
int main() {
       func();
}
```



```
x = 5, y = 5
x = 6, y = 5
x = 7, y = 5
x = 8, y = 5
x = 9, y = 5
```

Exercise 2

Write a program that will contain 2 structures: ***student*** and ***exam_day***. The first structure should contain information about the student's *name*, *surname*, *groupNo* and a variable for the second structure. The second structure should contain the *day*, *year* and *month* of the exam. The month has to be in letter representation (For example, May), not numbers.

Note: The program should require the user to enter all the fields for a student and its exam using the console, and then print them.

Exercise 2: Solution

```
#include <stdio.h>
struct exam_day
{
    int day;
    char month[9];
    int year;
};
struct student
{
    char name[15];
    char surname[15];
    int groupNo;
    struct exam_day exam;
};
```

```
int main() {
    struct student s;
    printf("Enter the name: ");
    scanf("%s", s.name);
    printf("\nEnter the surname: ");
    scanf("%s", s.surname);
    printf("\nEnter the group NO: ");
    scanf("%d", &s.groupNo);
    printf("\nEnter the day of exam: ");
    scanf("%d", &s.exam.day);
    printf("\nEnter the month of exam: ");
    scanf("%s", s.exam.month);
    printf("\nEnter the year of exam: ");
    scanf("%d", &s.exam.year);
printf("\nYou Entered: %s %s, his group is %d, the exam date is %d %s %d\n",
s.name, s.surname, s.groupNo, s.exam.day, s.exam.month, s.exam.year);
    getchar();
    return 0;
}
```

Exercise 3

Using a union, write a program that will read an **unsigned long long integer** via console and then encrypt it swapping values of each odd byte and its neighbour even byte, beginning with the most significant byte. The program must contain encryption(…) function

Standard output should contain 3 strings:

Original message: *xxx*

Encrypted message: *yyy*

Decrypted message: *xxx*

Exercise 3: Solution (1/3)

```
#include <stdio.h>
union messageStorage {
    unsigned char segments[8];
    unsigned long long int number;
};
```

Exercise 3: Solution (2/3)

```
// Encryption of an original message with replacing neighbour
even and odd bytes and further decryption of it
void encryption(union messageStorage ms) {
   char temp;
// first time for encryption and the second time for decryption
   for (int j = 0; j < 2; j++) {
       for (int i = 0; i < sizeof(ms.number); i=i+2) {
          temp = ms.segments[i];
          ms.segments[i] = ms.segments[i+1];
          ms.segments[i+1] = temp;
      }
      if (j==0) printf("Encrypted message: %lld \n", ms.number);
      else printf("Decrypted message: %lld \n", ms.number);
      getchar();
   }
}
```

Exercise 3: Solution (3/3)

```
int main() {
    char temp;
    union messageStorage ms;
    printf("size of t is %lu \n", sizeof(ms.number));
    printf("size of p is %lu \n", sizeof(ms.segments));
    printf("Enter the number: ");
    scanf("%lld", &ms.number);
    printf("Original message: %lld\n", ms.number);
    encryption(ms);
    return 0;
}
```

Exercise 4

Using a structure with bit fields, pack your *day*, *month* and *year* of birth into 2 bytes, considering that all fields consist of numbers.

Initialize numbers inside the code, print structure fields values in the console. Print the size of the structure in the console

Exercise 4: Solution

```
#include <stdio.h>
#define BASE_YEAR 1900
struct date
{
    unsigned short day : 5;
    unsigned short month : 4;
    unsigned short year : 7;
};
```

```
int main() {
    struct date birthday;
    birthday.day = 16;
    birthday.month = 12;
    birthday.year = 2000 - BASE_YEAR;          // birthday.year = 100
    printf("\n My birthday is %u.%u.%u \n", birthday.day, birthday.month,
birthday.year + BASE_YEAR);
    printf("\n Size of birthday structure is %lu bytes", sizeof(birthday));
    getchar();
    return 0;
}
```

Exercise 5

Write a program with array of structures for cookbook with recipes. Each recipe should contain its name, [2;10] ingredient, with the names and amount of those ingredients.

In the output, all the cookbook with recipes should be printed.

Exercise 5: Solution (1/4)

```
#include <stdio.h>
#define RECIPE_AMOUNT 3
#define MIN_INGREDIENTS 2
#define MAX_INGREDIENTS 10
struct cookbook
{
    char title[15];        // recipe title
    int ingredientsAmount;    // total amount of all ingredients
    char ingredient[MAX_INGREDIENTS][15];  // ingredient name
    int amount[MAX_INGREDIENTS];    // amount of a specific ingredient
};
```

Exercise 5: Solution (2/4)

```
int main() {
    struct cookbook recipe[RECIPE_AMOUNT];
    //inputting recipes to cookbook via Console
    for (int recipeCounter = 0; recipeCounter < RECIPE_AMOUNT; recipeCounter++) {
        printf("Recipe %d name is ", recipeCounter + 1);
        scanf("%s", recipe[recipeCounter].title);
        printf("Amount of ingredients for %s is ", recipe[recipeCounter].title);
        scanf("%d", &recipe[recipeCounter].ingredientsAmount);
        while (recipe[recipeCounter].ingredientsAmount < MIN_INGREDIENTS ||
        recipe[recipeCounter].ingredientsAmount > MAX_INGREDIENTS) {
            printf("Wrong number of ingredients, please use another amount: ");
            scanf("%d", &recipe[recipeCounter].ingredientsAmount);
        }
```

Exercise 5: Solution (3/4)

```
for (int ingredientCounter = 0; ingredientCounter < recipe[recipeCounter].ingredientsAmount;
ingredientCounter++) {
            printf("Ingredient %d name is ", ingredientCounter + 1);
            scanf("%s", recipe[recipeCounter].ingredient[ingredientCounter]);
            printf("Amount for ingredient %s is ", recipe[recipeCounter].ingredient[ingredientCounter]);
            scanf("%d", &recipe[recipeCounter].amount[ingredientCounter]);
            getchar();
        }
    }
```

Exercise 5: Solution (4/4)

```
 //printing cookbook to Console
    for (int recipeCounter = 0; recipeCounter < RECIPE_AMOUNT; recipeCounter++) {
        printf("\n%d. %s ", recipeCounter+1, recipe[recipeCounter].title);
        for (int ingredientCounter = 0; ingredientCounter < recipe[recipeCounter].ingredientsAmount;
ingredientCounter++) {
              printf("\n   %d.%d %s - %d ", recipeCounter+1, ingredientCounter+1,
              recipe[recipeCounter].ingredient[ingredientCounter],
              recipe[recipeCounter].amount[ingredientCounter]);
        }
    }
    getchar();
    return 0;
}
```