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

Structured programming

Exercises 2

Version 1.0, 22 September, 2016

Table of Contents

1. Simple C program structure

All the source code written in C is organized in **functions**

```
int main() {
  variable_declarations;
  expressions;
}
```

2. C Functions

- `main` - C **main** function
- In parentheses `()` we put the input arguments
- The return type of the functions is before the name (`int` – the function returns integer)
- The function body starts with `{`, and ends with `}`
- All declarations and expressions are separated with `;`

3. Comments usage

Comments are used for extra explanation or documenting the source code. C supports two types of comments:

- one line comments that start with double slash `//`

```
// comment in one line
```

- multiple line comments that can span across multiple lines and start with `/*` and end with `*/`

```
/*
Longer comment
in multiple
lines
*/
```

3.1. Example (Hello world)

Example 1

```
#include <stdio.h>

// main function
int main() {
    /*
    Printing a message on the standard output (the screen)
    */
    printf("Welcome to FINKI!\n");
    return 0;
}
```

- `#include` - directive for including external libraries
- `stdio.h` - library for accessing standard input/output streams (keyboard/screen)
- `printf` - function for printing on the standard output (screen)

3.2. Program for summing two integers

Example 2

```
#include <stdio.h>

int main() {
    int a = 5;
    int b = 10;
    int c = a + b;
    return 0;
}
```

4. Variables

- Variables are symbolic names for places in memory where are stored some values.
- Before using it, each variable must be **declared**.
- With each assignment of new value, the old value in the variable is lost.

4.1. Declaring variables

```
data_type variable_name = initial_value;
```

Example

```
int a = 5;
float x = 2.3;
char c = 'a';
```

4.2. Data types in C

Integers	Characters	Real numbers
`int`	`char`	`float`
`short`		`double`
`long`		

4.3. Defining variable names

In naming variables you can use:

- lowercase letters from `a` to `z`;
- uppercase letters from `A` to `Z`;
- digits from `0` to `9` (the name can not start with digit);

When choosing a variable name, pick ones that clearly describe the value they store.

C is **case sensitive**

5. Named constants

Names constants are created using the keyword `const`

Example 3

```
#include <stdio.h>

int main() {
    const long double PI = 3.141592653590L;
    const int DAYS_IN_WEEK = 7;
    const SUNDAY = 0; // by default int
    DAYS_IN_WEEK = 7; // error
    return 0;
}
```

Named constants can be created also by using the preprocessor and with all uppercase letters by convention.

Using `#define`

```
#define TEXT_TO_SEARCH_FOR REPLACEMENT_TEXT
```

Example 4

```
#include <stdio.h>
#define PI 3.141592653590L
#define DAYS_IN_WEEKS 7
#define SUNDAY 0

int main() {
    long number = PI;
    int day = SUNDAY;
    return 0;
}
```

6. Printing on the standard output

For printing on the standard output (screen) in C we use the function `printf` from the library `stdio.h` (**St**andar**d** **I**nput **O**tput)

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

The signature of the function is:

```
int printf(control_array, list_of_arguments)
```

The *control array* contains text of any kind, and format placeholders with leading `%` or special characters with leading `\`.

The *format placeholders* are determined from the variable type we want to print.

6.1. Most used format placeholders

Format	Usage
`%d`	integers (`int`)
`%i`	integers (`int`)
`%f`	real numbers (`float`, `double`)
`%c`	characters (`char`)
`%s`	characters (стринг, `char[]`, `char*`)
`%%`	character `%`

6.2. Usage of function `printf`

Print on the standard output the following sentences:

```
First sentence.
Second sentence.
Third sentence.
```

Example 5

```
#include <stdio.h>

int main() {
    printf("First sentence.\n");
    printf("Second sentence.\nThird sentence.\n");
    return 0;
}
```

Пример 6

```
#include <stdio.h>

int main() {
    printf(" is a word long %d letters.\n", printf("Macedonia"));
    return 0;
}
```

7. Operators

7.1. Arithmetic operators

Are used on numbers (integers or real)

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo (residue after division)

8. Problems

8.1. Problem 1

Write a program that will compute the value of the mathematical expression: `x = 3/2 + (5 - 46*5/12)`

Solution 1

```
#include <stdio.h>

int main() {
    float x = 3.0 / 2 + (5 - 46 * 5.0 / 12);
    printf("x = %.2f\n", x);
    return 0;
}
```

8.2. Problem 2

Write a program that for given value of x (during the declaration) will compute and print the value of x^2.

Solution 2

```
#include <stdio.h>

int main() {
    int x = 7;
    printf("Number %d squared is %d\n", x, x * x);
    return 0;
}
```

8.3. Problem 3

Write a program that for a given sides of one triangle, it will print the perimeter and area of the square (values are `a = 5`, `b = 7.5`, `c = 10.2`).

Solution 3

```
#include <stdio.h>

int main() {
    float a = 5.0;
    float b = 7.5;
    float c = 10.2;
    float L = a + b + c;
    float s = L / 2;
    float P = s * (s - a) * (s - b) * (s - c);
    printf("Perimeter is: %.2f\n", L);
    printf("Area is: %.2f\n", P);
    return 0;
}
```

8.4. Problem 4

Write a program for computing the arithmetic mean of the numbers 3, 5 and 12..

Solution 4

```c
#include <stdio.h>

int main() {
    int a = 3;
    int b = 5;
    int c = 12;
    float as = (a + b + c) / 3.0;
    printf("The arithmetic mean is %2.f\n", as);
    return 0;
}
```

8.5. Problem 5

Write a program that will print the remainder from the division of number 19 with 2, 3, 5 and 8.

Solution 5

```c
#include <stdio.h>

int main() {
    int a = 19;
    printf("The residue of division with 2 is: %d\n", a % 2);
    printf("The residue of division with 3 is: %d\n", a % 3);
    printf("The residue of division with 5 is: %d\n", a % 5);
    printf("The residue of division with 8 is: %d\n", a % 8);
    return 0;
}
```

8.6. Problem 6

Write a program for computing and printing the circle area and perimeter. The circle radius is read as decimal number.

Solution 6

```c
#include <stdio.h>
#define PI 3.14

int main() {
    float radius;
    scanf("%f", &radius);

    float perimeter = 2 * radius * PI ;
    float area = radius * radius * PI;
    printf("L = %f\n", perimeter);
    printf("P = %f\n", area);
    return 0;
}
```

8.7. Problem 7

Write a program that reads from standard input two integers and prints their sum, difference, product and division remainder.

Solution 7

```
#include <stdio.h>

int main() {
    int x, y;
    scanf("%d %d", &x, &y);

    printf("%d + %d = %d\n", x, y, x + y);
    printf("%d - %d = %d\n", x, y, x - y);
    printf("%d * %d = %d\n", x, y, x * y);
    printf("%d %% %d = %d\n", x, y, x % y);
    return 0;
}
```

8.8. Problem 8

Write a program that reads uppercase letter from standard input and prints out in lowercase.

Each character is represented with its ASCII code.

example

```
A = 65, a = 97
```

Solution 8

```
#include <stdio.h>

int main() {
    char c;
    printf("Enter an uppercase letter: ");
    scanf("%c", &c);
    printf("%c lowercase is: '%c'\n", c, c + ('a' - 'A'));
    return 0;
}
```

9. Source code of the examples and problems

https://github.com/finki-mk/SP/

Source code ZIP