



"Ss. Cyril and Methodius" University in Skopje

**FACULTY OF COMPUTER
SCIENCE AND ENGINEERING**

Structured programming

Exercises 2

Верзија 1.0, 22 September, 2016

Содержина

| | |
|--|---|
| 1. Simple C program structure..... | 1 |
| 2. C Functions..... | 1 |
| 3. Comments usage..... | 1 |
| 3.1. Example (Hello world) | 1 |
| 4. Named constants | 2 |
| 5. Arithmetic operators | 3 |
| 6. Printing format placeholders | 3 |
| 6.1. Usage of function printf | 3 |
| 7. Problems..... | 4 |
| 7.1. Problem 1..... | 4 |
| 7.2. Problem 2..... | 4 |
| 7.3. Problem 3..... | 4 |
| 7.4. Problem 4..... | 5 |
| 7.5. Problem 5..... | 5 |
| 7.6. Problem 6..... | 5 |
| 7.7. Problem 7..... | 6 |
| 7.8. Problem 8..... | 6 |
| 8. Source code of the examples and problems..... | 8 |

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)

Structured programming

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)

===

Example 2

```
#include <stdio.h>

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

4. 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;
}
```

5. Arithmetic operators

Applied on numbers (integers or real)

| Operator | Operation |
|----------|--|
| + | Addition |
| - | Subtraction |
| * | Multiplication |
| / | Division |
| % | Division by modulo (residue of the division) |

6. Printing format placeholders

| Placeholder | Usage |
|-------------|---|
| %d | integers (int) |
| %i | integers (int) |
| %f | real numbers (float, double) |
| %c | characters (char) |
| %s | array of characters (strings char[], char*) |

6.1. Usage of function printf

Print on the standard output the following sentences:

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

Example 3

```
#include <stdio.h>

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

7. Problems

7.1. Problem 1

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

Solution 1

```
#include <stdio.h>

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

7.2. 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;
}
```

7.3. 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;
}
```

7.4. Problem 4

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

Solution 4

```
#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 %.2f\n", as);
    return 0;
}
```

7.5. Problem 5

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

Solution 5

```
#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;
}
```

7.6. Problem 6

Write a program for computing and printing the circle area and perimeter. The circle

radius is read as decimal number.

Solution 6

```
#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;
}
```

7.7. 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;
}
```

7.8. 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;
}
```

8. Source code of the examples and problems

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

Source code ZIP