Structured programming

Exercises 4

Table of Contents

1. Flow control statements `if-else`

1.1. From lectures
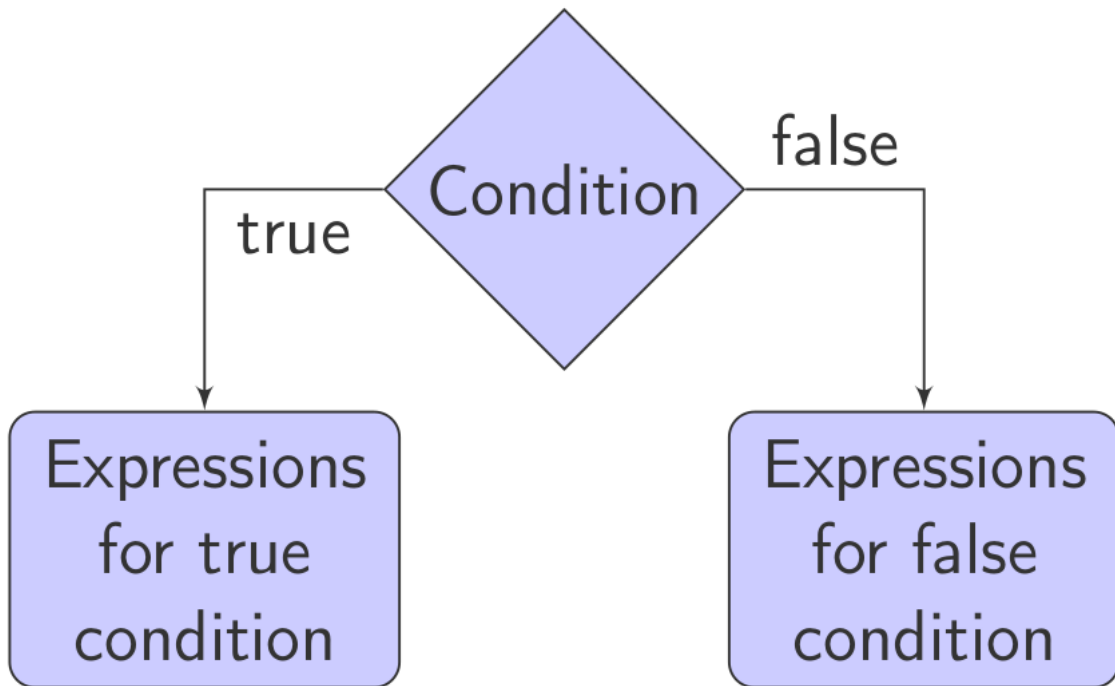

```
if ( condition ) {
    expressions_for_true_condition;
}
else {
    expressions_for_false_condition;
}
```

1.2. What will be the output?

Example `ex4_1_en.c`

```c
#include <stdio.h>
int main() {
    int m = 5, n = 10;
    if (m > n)
        ++m;
    ++n;
    printf("m = %d, n = %d\n", m, n);
    return 0;
}
```

Output

```
m = 5, n = 11
```

2. Problems

2.1. Problem 1

Write a program that will print the maximum of two number read from SI.

Solution `p4_1a.c`

```c
#include <stdio.h>

int main() {
    int a, b;
    printf("Vnesi 2 broja: \n");
    scanf("%d %d", &a, &b);
    if (a > b)
        printf("Maximum: %d\n", a);
    else
        printf("Maximum: %d\n", b);
    return 0;
}
```

Solution b – without if-else `p4_1b.c`

```c
#include <stdio.h>
int main() {
    int a, b;
    printf("Vnesi 2 broja: \n");
    scanf("%d %d", &a, &b);
    printf("Maximum: %d\n", (a > b) ? a : b);
    return 0;
}
```

2.2. Problem 2

Write a program that checks if given year read from SI is leap or not and prints out a appropriate message.
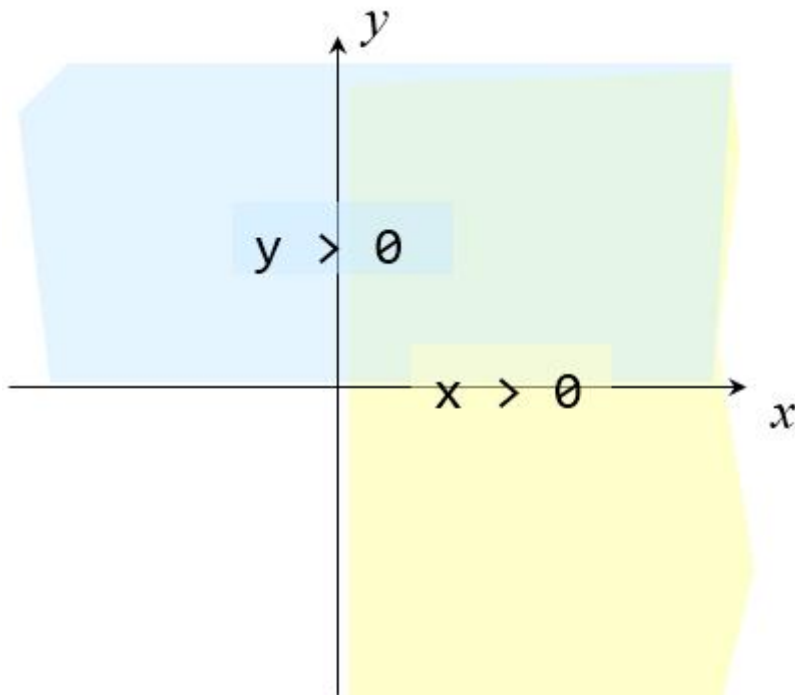
Example leap years

1976, 2000, 2004, 2008, 2012…

> The year is leap is divisible by 4 and not divisible by 100, or divisible by 400

Solution `p4_2_en.c`

```c
#include <stdio.h>

int main() {
    int year;
    printf ("Enter year: \n");
    scanf("%d", &year);
    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
        printf("%d is leap.\n", year);
    } else {
        printf("%d is not leap.\n", year);
    }
    return 0;
}
```

2.3. Problem 3

The coordinates of a point are read from SI. Write a program that will print out the quadrant or the axis where the point belongs. If the point lays on the origin, print out a appropriate message.
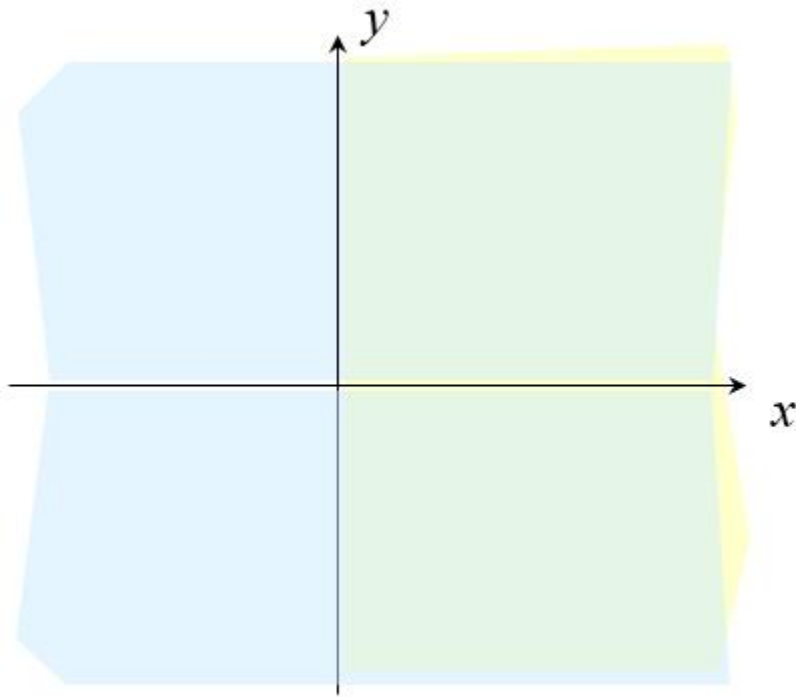
Solution (a) `p4_3a_en.c`

```c
#include <stdio.h>

int main () {
    float x, y;
    printf ("Enter coordinates \n");
    scanf ("%f %f", &x, &y);
    if (x > 0 && y > 0)
        printf("I quadrant.\n");
    if (x > 0 && y < 0)
        printf("IV quadrant.\n");
    if (x < 0 && y > 0)
        printf("II quadrant.\n");
    if (x < 0 && y < 0)
        printf("III quadrant.\n");
    return 0;
}
```


Figure 1. Solution (b) `p4_3b_en.c`

```c
#include <stdio.h>
int main () {
    float x, y;
    printf("Enter coordinates \n");
    scanf("%f %f", &x, &y);
    if (x > 0)
        if (y > 0)
            printf("I quadrant.\n");
        else
            printf("IV quadrant.\n");
    else if (y > 0)
        printf("II quadrant.\n");
    else
        printf("III quadrant.\n");
    return 0;
}
```

1. *Does this program cover all the cases?*

2. *Will it print anything for any given point?*

```c
#include <stdio.h>
int main () {
    float x, y;
    printf ("Enter coordinates \n");
    scanf ("%f %f", &x, &y);
    if (x > 0)
        if (y > 0)
            printf("I quadrant.\n");
        else if (y < 0)
            printf("IV quadrant.\n");
        else
            printf("Positive X axis.\n");
    else if (x < 0)
        if (y > 0)
            printf("II quadrant.\n");
        else if (y < 0)
            printf("III quadrant.\n");
        else
            printf("Negative X axis.\n");
    else if (y > 0)
        printf("Positive Y axis.\n");
    else if (y < 0)
        printf("Negative Y oska.\n");
    else
        printf("Origin.\n");
    return 0;
}
```

2.4. Problem 4

Write a program that will generate and print the grade according to the following table:

Points	Grade
0-50	5
51-60	6
61-70	7
71-80	8
81-90	9
91-100	10

```c
#include <stdio.h>

int main () {
    int points, grade = 0;
    printf("Enter points: \n");
    scanf("%d", &points);
    if (points >= 0 && points <= 50) grade = 5;
    else if (points > 50 && points <= 60) grade = 6;
    else if (points > 60 && points <= 70) grade = 7;
    else if (points > 70 && points <= 80) grade = 8;
    else if (points > 80 && points <= 90) grade = 9;
    else if (points > 90 && points <= 100) grade = 10;
    else printf("Invalid value for points!\n");
    printf("Grade %d\n", grade);
    return 0;
}
```

```c
#include <stdio.h>

int main () {
    int points, grade = 0;
    printf("Enter points: \n");
    scanf("%d", &points);
    if (points < 0 || points > 100)
        printf("Invalid value for points!\n");
    else {
        if (points > 90) grade = 10;
        else if (points > 80) grade = 9;
        else if (points > 70) grade = 8;
        else if (points > 60) grade = 7;
        else if (points > 50) grade = 6;
        else grade = 5;
        printf("Grade %d\n", grade);
    }
    return 0;
}
```

2.5. Problem 5

Change the previous program, so the sign of the number should be printed (+/-) depending on the last digit of the points number:

last digit	prints
1 - 3	-
4 - 7	<blank_space>
8 - 0	+

example

```
81 = 9-
94 = 10
68 = 7+
```

ℹ️ For grade 5 doesn't add + or –, and for grade 10 should not add +.

Solution `p4_5_en.c`

```c
#include <stdio.h>

int main () {
    int points, grade = 0;
    printf("Enter points: \n");
    scanf("%d", &points);
    if (points < 0 || points > 100)
        printf("Invalid value for points!\n");
    else {
        if (points > 90) grade = 10;
        else if (points > 80) grade = 9;
        else if (points > 70) grade = 8;
        else if (points > 60) grade = 7;
        else if (points > 50) grade = 6;
        else grade = 5;
        char sign = ' ';
        int pc = points % 10;
        if (grade != 5) {
            if (pc >= 1 && pc <= 3) sign = '-';
            else if (grade != 10 && (pc >= 8 || pc == 0))
                sign = '+';
        }
        printf("Grade %d%c\n", grade, sign);
    }
    return 0;
}
```

2.6. Problem 6

Read from standard input three numbers in arbitrary order. The numbers are lengths of triangle sides. Write a program that will check if triangle can be constructed from given lengths, if so, then should check if the triangle is right triangle and compute its area. On contrary, appropriate messages should be printed.

Solution `p4_6_en.c`

```c
#include <stdio.h>

int main() {
    float a, b, c;
    printf("Enter lengths of sides: \n");
    scanf("%f %f %f", &a, &b, &c);
    if ((a + b <= c) || (a + c <= b) || (b + c <= a))
        printf("A triangle can not be constructed.\n");
    else {
        if (a >= b) {
            float tmp = a;
            a = b;
            b = tmp;
        }
        if (a >= c) {
            float tmp = a;
            a = c;
            c = tmp;
        }
        if (b >= c) {
            float tmp = b;
            b = c;
            c = tmp;
        }
        // now the longest side is in the variable c
        if (c * c == a * a + b * b) {
            printf("The triangle is right triangle.\n");
            printf("Area is %7.3f\n", a * b / 2);
        } else {
            printf("The triangle is NOT a right triangle.\n");
        }
    }
    return 0;
}
```

2.7. Problem 7

Write a program for simple calculator. The program reads two numbers and operator in format:

`num1 operator num2`

After the operation, depending on the operator, the result should be printed in format:

```
num1 operator num2 = result
```

Solution `p4_7_en.c`

```c
#include <stdio.h>
int main() {
    char op;
    float num1, num2, result;
    printf("Enter two numbers and operator in format\n");
    printf(" num1 operator num2\n");
    scanf("%f %c %f", &num1, &op, &num2);
    if (op == '*')  {
        result = num1 * num2;
    }
    else if (op == '+')  {
        result = num1 + num2;
    }
    else if (op == '-') {
        result = num1 - num2;
    }
    else if (op == '/') {
        if (num2) {
            result = num1 / num2;
        }
        else {
            printf("Division by 0!\n");
            return 0;
        }
    } else {
        printf("Invalid operator!\n");
        return 0;
    }
    printf("%f %c %f = %f\n", num1, op, num2, result);
    return 0;
}
```

3. Try these problems at home

3.1. Problem 1

For three segments read from SI, the program should check if a triangle can be formed, then print the kind.

3.2. Problem 2 *

For given center of circle and it's radius the program should determine the quadrants it is overlapping.

4. Source code of the examples and problems

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

Source code ZIP