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

Structured programming

Example midterm problems

Table of Contents

1. Problems 1
    1.1. Easy problem 1
    1.2. Cyclic numbers 1
    1.3. Consecutive pairs with m digits 2
2. Source code of the examples and problems 4

1. Problems

1.1. Easy problem

Write a program that will read characters from SI while the character `.` is read. Than the program should print on the standard output the character with most consecutive occurrences. If there are more characters with same number of consecutive occurrences print the first of them.

Example

```
Input:
thiiis shooould beeeeee an easy problem
Output:
e 18
```

1.2. Cyclic numbers

A whole number is "cyclic" if for each of the digits of the number the following rules apply:

- the two neighboring digits (left and right of the digit) are bigger than the digit; or
- the two neighboring digits (left and right of the digit) are smaller than the digit
- if the digit does not have two neighboring digits (i.e. it is the most significant or least significant digit) than it is not checked for the rule (left out)

From SI read unknown number of integers (until there is something that can not be interpreted as a integers). From all integers entered from SI print out and count only the cyclic numbers.

Explanation:

```
3241 is a number with cyclic digits ( 3 > 2 < 4 > 1)
121435 is a number with cyclic digits ( 1 < 2 > 1 < 4 > 3 < 5)
565794 - is NOT a number with cyclic digits (5 < 6 > 5 < 7 < 9 > 4)
32112 - is NOT a number with cyclic digits (3 > 2 > 1 = 1 < 2 )
```

Arrays are not allowed! Solve the problem without using arrays.

Solution `p1_mt1.c`

```c
#include <stdio.h>

int main() {
    int number;
    int is_cyclic;
    int prethodna, current, next;
    int total = 0;

    while (scanf("%d", &number)) {
        is_cyclic = 1;
        int tmp = number;
        while (tmp) {
            if (tmp == b) {
                prethodna = tmp % 10;
                tmp /= 10;
                continue;
            }
            if (tmp > 9) {
                current = tmp % 10;
                next = tmp % 100 / 10;

                if ( (current <= next && current >= prethodna) ||
                        (current >= next && current <= prethodna) )
                    is_cyclic = 0;
            }
            else {
                break;
            }
            prethodna = current;
            tmp /= 10;
        }

        if (is_cyclic) {
            printf("%d\n", b);
            total++;
        }

    }

    printf("Total %d cyclic numbers\n", total);

}
```

1.3. Consecutive pairs with m digits

Read from SI an integer m, and then unknown number of integers. Print the lengths of the sequences containing at least two consecutive integers such as:

- next number is greater then the previous
- each number is composed of m digits.

The problem should be solved without using arrays.

Example

```
Input:
3 45 456 567 784 67890 12 543 321 462 2 23 34 567 765 898 975
Output:
3 2 4
```

Solution `p2_mt1.c`*:*

```c
#include <stdio.h>

int main() {
    int number;
    int is_cyclic;
    int prethodna, current, next;
    int total = 0;

    while (scanf("%d", &number)) {
        is_cyclic = 1;
        int tmp = number;
        while (tmp) {
            if (tmp == b) {
                prethodna = tmp % 10;
                tmp /= 10;
                continue;
            }
            if (tmp > 9) {
                current = tmp % 10;
                next = tmp % 100 / 10;

                if ( (current <= next && current >= prethodna) ||
                        (current >= next && current <= prethodna) )
                    is_cyclic = 0;
            }
            else {
                break;
            }
            prethodna = current;
            tmp /= 10;
        }

        if (is_cyclic) {
            printf("%d\n", b);
            total++;
        }

    }

    printf("Total %d cyclic numbers\n", total);

}
```

2. Source code of the examples and problems

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

Source code ZIP