FEEDBACK CV2

!!!!GLOBAL!!!!

- Dodrziavat format vypisov aky je v zadani, pokial je v zadani "vystup odriadkujte", treba dat nakoniec "\n"
- Pokial v zadani nie je ziaden vzorovy vstup/vystup, mozte si dat aky chcete, pokial je, treba dodrziavat zadanie
- PRI TESTOCH / PROJEKTOCH TREBA DODRZIAVAT PRESNE VYPISY ZADANIA - v opacnom pripade sa budu strhavat body
- Rozdiel medzi C90 a C99
    - rozdielna definicia for loop
        - C99 podporuje `for (int i = 0; i <= limit; i++)`
        - C90 podporuje `int i = 0; for(i = 0; i <= limit; i++)` -> C90 nepodporuje definovanie for loop tak ako C99 (C90 does not support defining for loop as shown in C99 method)

Uloha 3 / Task 3

- zbytocne pouzivanie premennych / if statements (unnecessary usage of variables, if statements)
- uplne stacilo spravit `printf("%d %d", int(number), (int)(number+0.5));`
- nemozeme spravit `(int)(number+1)`, lebo ked mame napriklad cislo 3.4, tak v tomto pripade by vysledok bol 4 a nie 3 (we cannot use `(int)(number+1)`, because if we take example number 3.4, the result would be 4 instead of 3)

Uloha 5 / Task 5

- Treba osetrit delenie 0, pokial nie je osetrene vyhodi error -> vzdy sa treba zamyslat nad edge-cases aj ked nie su priamo uvedene v zadani (It is necessary to handle division by 0, otherwise program ends with an error -> always consider as many edge-cases as possible even if they are not directly mentioned in task)

Uloha 7 / Task 7

- V pripade, ze mam vyraz:
    - `int a = 5; int b = 0; b = ++a;` -> hodnota a sa najprv inkrementuje a potom priradi, takze po vykonani posledneho riadku budu hodnoty **a=6, b=6**
    - `int a = 5; int b = 0; b = a++;` -> hodnota a sa najprv priradi a az nasledne inkrementuje, takze po vykonani riadku budu hodnoty **a=6, b=5**
- V priklade co ste mali `e / --a * b++ / c++`

```
Ked su pociatocne hodnoty a = 6, b = 2, c = 2, d = 4, e = 5

Tak tento vyraz sa vyhodnocuje zlava doprava nasledovne
e(5) / --a(5, najprv dekrementacia) * b++(2, najprv priradenie) / c++(2,
najprv priradenie)
```

```
Takze vyraz vyzera nasledovne: 5 / 5 * 2 / 2
Hodnoty premennych po vykonani vypoctu: a = 5, b = 3, c = 3, d = 4, e = 5
```

- V druhom priklade `a %= b = d = 1 + e / 2`

```
Ked su pociatocne hodnoty a = 6, b = 2, c = 2, d = 4, e = 5

Hodnoty sa vzdy priraduju sprava dolava (right to left), to znamena, ze
vyraz 1 + e / 2 sa priradi do premennej b

Vypocet je nasledovny: 1 + e / 2 => 1 + 5 / 2 = 3 (kedze robime s celymi
cislami, tak 5 / 2 = 2)

Vysledok vypoctu vyssie sa priradi do premennych d a b
Ako posledne sa vykona a % b (a modulo b) => zvysok po deleni, cize 6 % 3 =
0
```

Uloha 8 / Task 8

- neinicializovat min a max na 0, su rozne sposoby ako to spravit (do not inicialize min and max values to 0, there are other ways to inicialize them):
  - INT_MIN, INT_MAX -> #include <limits.h>
  - (+-)INFINITY -> #include <math.h>
  - (+-)DBL_MAX, (+-)FLT_MAX -> #include <float.h>
  - alebo inicializovat hodnoty min a max na prve cislo v sekvencii (or inicialize min and max to first number in sequence)

Uloha 9 / Task 9

- ak je v zadani napisane, ze treba pouzit pole, tak ho treba pouzit (if task states to use list / array, use it)
- da sa nacitavat aj priamo do array, netreba inicializovat novu premennu a potom priradovat hodnotu na index:
  - `scanf("%d", &array[i]);`
- neodporucam inicializovat pole s hodnotou zo scanf (i dont recommend inicializing arrays with value from scanf) -> this is only supported in C99 and above standard
  - `scanf("%d", &length); int pole[length];` -> NOT RECOMMENDED