Těžký příklad — Minesweeper

1 Zadání HW09

- Napište program **minesweeper.py**, který najde pozici min podle informace o počtu min v okolí každého políčka.
- **Vstup:** první argument programu na příkazové řádce je cesta a jméno k souboru s informací o počtu min (použijte `sys.argv[1]`):
  - Soubor obsahuje celá čísla oddělená mezerou. Čísla jsou v rozsahu 0 až 9.
  - Zadané herní pole je obdélníková matice, všechny řádky mají stejný počet čísel.
  - Číslo znamená kolik min se nachází na políčku a v jeho okolí. Minimum je 0, žádná mina na políčku ani v jeho okolí. Maximum je 9, mina je na políčku a i na každém políčku v okolí.
  - Pro políčko v rohu existují pouze 3 pole v okolí. Pro pole na okraji existuje 5 políček v okolí. Pro pole uvnitř existuje 8 políček v okolí.
  - Příklad vstupního souboru

```
1 2 1 1 0 1 1
2 3 1 1 0 1 1
3 4 3 2 1 1 1
2 2 2 1 1 0 0
1 1 3 3 3 1 0
0 0 1 2 2 1 0
```

- **Výstup:** Vytištěné pole znaků `.` a `X` o stejných rozměrech jako byl vstup:
  - `.` pokud na dané pozici není mina
  - `X` pokud na dané pozici je mina
  - Příklad výstupu

```
.......
X.X...X
X......
.X.X...
.......
...XX..
```

1.1 Poznámky

- Předpokládejte, že všechny vstupy jsou zadány správně, soubor s názvem `sys.argv[1]` existuje a obsahuje zadané herní pole v uvedeném formátu.
- Můžete také předpokládat, že pro vstup existuje řešení.
- Snažte se napsat rychlý algoritmus, limit pro uvedené příklady je 15 s.
  - Pokud je v souboru 0 pak víte, že žádné okolní políčko neobsahuje minu.

- Pokud je v rohu číslo 4, nebo na straně číslo 6, nebo uprostřed číslo 9, pak jsou všude miny i v místě čísla.
- Jinak zkuste položit minu a aplikovat výše uvedená pravidla s tím, zda označený počet min odpovídá zadanému číslu, nebo neoznačená místa odpovídají počtu min, která zbývá označit do zadaného čísla.

Program v souboru **minesweeper.py** odevzdejte pomocí odevzdávacího systému (úloha HW09).

2 Příklady

Níže uvedené příklady lze stáhnout ze stránek cvičení.

2.1 test1.txt

Vstup programu je soubor test1.txt, který obsahuje:

```
1 2 1 1 0 1 1
2 3 1 1 0 1 1
3 4 3 2 1 1 1
2 2 2 1 1 0 0
1 1 3 3 3 1 0
0 0 1 2 2 1 0
```

Výstup programu bude:

```
.......
X.X...X
X......
.X.X...
.......
...XX..
```

2.2 test2.txt

Vstup programu je soubor test2.txt, který obsahuje:

```
2 4 4 3 2 2 4 4 5 5 5 4 3 3 4 5 4
3 6 7 6 5 4 6 6 7 7 7 6 4 3 5 6 5
3 6 7 7 7 6 7 6 7 6 5 4 2 2 4 6 5
4 7 8 8 8 7 6 6 7 7 6 5 4 3 5 6 5
4 7 7 7 7 6 6 6 7 7 6 6 6 5 5 6 5
4 7 8 7 7 5 6 6 6 6 6 8 9 8 7 7 5
3 5 7 5 6 4 6 6 6 6 6 7 8 7 7 7 5
3 5 7 6 7 6 6 6 6 7 6 5 6 7 9 9 6
4 5 6 5 7 7 7 6 6 7 7 5 6 7 9 9 6
5 7 7 7 8 8 7 6 6 6 5 3 5 6 8 8 6
5 7 8 7 8 8 7 5 4 4 5 4 6 5 6 6 5
4 7 8 8 7 7 5 4 3 5 6 5 6 5 5 5 4
4 7 7 7 6 7 6 4 3 4 7 7 8 7 5 5 3
3 5 4 4 3 4 4 3 3 3 5 5 6 6 4 4 2
```

Výstup programu bude:

```
.XX....XXXXX.X.XX
X.XX.XXX.XX.X.XXX
.XXXXX.XX.XX...X.
.XX.XX.XXX.....XX
XXXXXXX.XXXXXXXXX
X.XX.X.XX.XXXX..X
.XXX.X.XX..XXXXXX
.X.X.XX.XXXX.XXXX
X.XXXXX.XXX..XXXX
XX.X.XXX.X.XXXXXX
XXXXXXX.XX...X.XX
.XXXXXX...XX.X..X
X.XX..X..XXXXXX.X
XXX.XXXX.X.XXXX.X
```