

Cvičení 9 – skupiny IB113/06,07

2D seznamy a hry

Úkoly s maticemi

- vypsat diagonálu matice A (matice má tvar seznamu seznamů, zanořené seznamy jsou řádky – např. `[[1, 10], [9, -5]]` je matice, která má na diagonále čísla 1 a -5.

```
print_diagonal(A)
```

- najít sloupeček matice A s největším součtem

```
column_with_max_sum(A) ... i
```

- najít souřadnice největšího prvku

```
coordinates_of_max(A) ... (i, j)
```

- spočítat počet nul v matici A

```
count_of_zeros(A) ... m
```

- transpozice matice A

```
transpose(A) ... B = AT
```

- součet matic A, B

```
sum_matrices(A, B) ... C = A + B
```

- součin matic A, B těch správných rozměrů (pokud znáte z matematiky)

```
product_matrices(A, B) ... C = AB
```

- zjistit, zda se někde v matici vyskytují dvě stejná čísla vedle sebe (v řádku nebo sloupci)

```
equal_neighbours_exist(A) ... True / False
```

2D game

- Přečtěte si kód na další stránce
- Vyzkoušejte v Thonny, co to dělá
https://is.muni.cz/auth/el/fi/podzim2023/IB113/um/skupiny_06_07/cviceni_9/tic_tac_toe.py
- Co by se dalo vylepšit?
- Zkuste vylepšit:)

```

N = 3
def empty_plan():
    return [[0 for i in range(N)] for j in range(N)]

def determine_winner(plan):
    for i in range(N):
        all_same = True
        for j in range(N):
            if plan[i][j] != plan[i][0]:
                all_same = False
        if all_same and plan[i][0] != 0:
            return plan[i][0]

    all_same = True
    for j in range(N):
        if plan[j][i] != plan[0][i]:
            all_same = False
    if all_same and plan[0][i] != 0:
        return plan[0][i]
    return 0

def print_plan(plan):
    symbol = {0: ".", 1: "X", 2: "O"}
    for i in range(N):
        for j in range(N):
            print(symbol[plan[i][j]], end=" ")
        print()

def play():
    plan = empty_plan()
    player = 1
    while determine_winner(plan) == 0:
        print_plan(plan)
        move = input("Player "+str(player)+" move:")
        x, y = list(map(int, move.split(" ")))
        plan[y-1][x-1] = player
        player = 3 - player
    print_plan(plan)
    print("Player "+str(determine_winner(plan))+" wins.")

play()

```

Pro rychlíky (máte-li všechno z předchozích stránek hotovo):

https://www.fi.muni.cz/~xpelanek/IB113/sbirka/08-datove_struktury.html

- 8.5.1 – Vyhodnocení logického výrazu v postfixové notaci (využijete datovou strukturu zásobník – 8.1)
- 8.5.3 – Kontrola uzávorkování (také se hodí zásobník)