Semestrální práce č.15

Tomáš Janovec

December 20, 2022

1 Specifikace požadavků

1.1 Zadání práce

Zapište program, který do obdélníkové matice zadané velikosti zapíše čísla 1..n*m po spirále – počínaje levým horním rohem“, ke středu, ve směru hodinových ručiček. Hodnoty n a m představují počet řádků a počet sloupců vytvářené matice.

Specifikace vstupu: Program má umožnit při jednom spuštění zpracování libovolného počtu zadání. Program bude postupně načítat dvojice čísel jako velikost vytvářené matice. Pro každé zadání nechť program vypíše výslednou matici. Po načtení záporného nebo nulového čísla namísto prvního rozměru matice nechť program skončí svoji činnost.

1.2 Ukázka komunikace programu s uživatelem:

Pocet radku
2
Pocet sloupcu
2
Vysledna matice
1 2
4 3
Pocet radku
4
Pocet sloupcu
3
Vysledna matice
1 2 3
10 11 4
9 12 5
8 7 6
Pocet radku
4
Pocet sloupcu
6
Vysledna matice
1 2 3 4 5 6
16 17 18 19 20 7
15 24 23 22 21 8
14 13 12 11 10 9
Pocet radku
-1

2 Návrh řešení

2.1 Stručný popis algoritmu

Rozhodl jsem se postupovat následovně. Postupně bude vyplňována matice inkrementujícími hodnotami ve směru spirály. Tedy nejdříve vyplním řádek zleva doprava, sloupec shora dolů, řádek zprava doleva a sloupec zdola nahoru. Pozice, od které metoda bude zapisovat inkrementovanou hodnotu v daném slupci/řádku, se vezme vždy od poslední přepsané pozice v matici. Pro zápis inkrementující se hodnoty do matice, daná souřadnice v matici vždy musí plnit podmínku, že hodnota na jejím místě je nula a zároveň není mimo rozsah matice.

Toto se bude opakovat, dokud nebude inkrementující se hodnota větší rovna hodnotě m*n.

2.2 Implementace

Co se kódu týče, matici, do které se budou inkrementované hodnoty úkladát, bude reprezentována dvojrozměrným polem, které má na začátku samé nuly. Před začátkem cyklu, se na souřadnici [0,0] uloží jednička (je to nutné, jelikož metoda vždy nejdříve udělá krok a až poté zapíše hodnotu). Aktuální souřadnice pro zápis hodnoty a inkrementující se hodnota budou uložená v poli coord [x,y,hodnota].

Kód je rozčleněn na čtyři void podmetody, z nichž každá obsluhuje jeden směr. Rozhodl jsem se tak udělat z důvodu přehlednosti a pohodlnosti testovaní (lze tak otestovat každou metodu pohodlně zvlášť). Také to znamená, že se kód dá využít na tvorbu spirální matice v opačném směru. Je však nutné poznamenat, že kód se dá relativně přehledně napsat pouze do jedné metody.

Každá taková metoda bude nést pojmenování ve tvaru go(směr). Každá z těchto metod bude mít argumenty matici a pole coord. Metoda bude přepisovat samotnou matici a bude i přepisovat aktuální souřadnice a inkrementující hodnotu (pole coord).

Všechny podmetody go, budou dále použity v hlavní metodě makeArray, která podmetody spojí do požadovaného cyklu. Metoda dostane jako vstup počet řádků a sloupců, z nichž vytvoří danou matici, kterou bude pak společně s polem coord předávat podmetodám go. Výstupem této metody je požadovaná spirální matice.

Pro celkové splnění zadání, které požaduje možnost tvorby více spirálních matic, obsahuje kód metodu start, která načte vstup od uživatele, ověří zdali je validní a bude vypisovat spirální matici dokud uživatel nezadá koncový vstup. V případě nevalidního vstupu při zadání záporného počtu sloupců, se program bude dotazovat uživatele, dokud nedostane validní vstup.

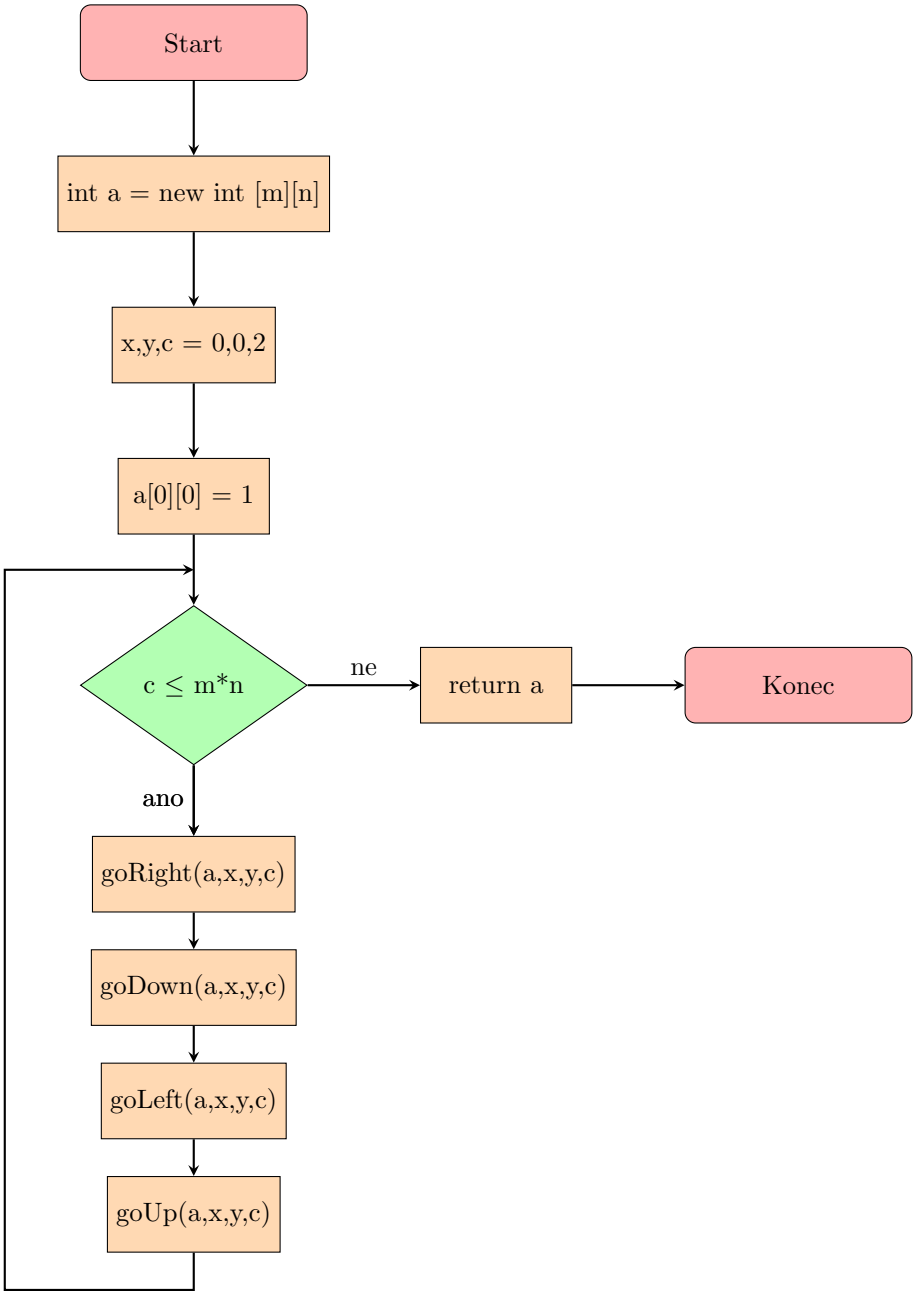
2.3 Časová složitost

Časová složitost tohoto algoritmu se dá spočítat jednoduše. Vímeli, že každý prvek matice navštívíme a zapíšeme do něj právě jednou, můžeme říct, že taková časová složitost bude $O(m * n)$.

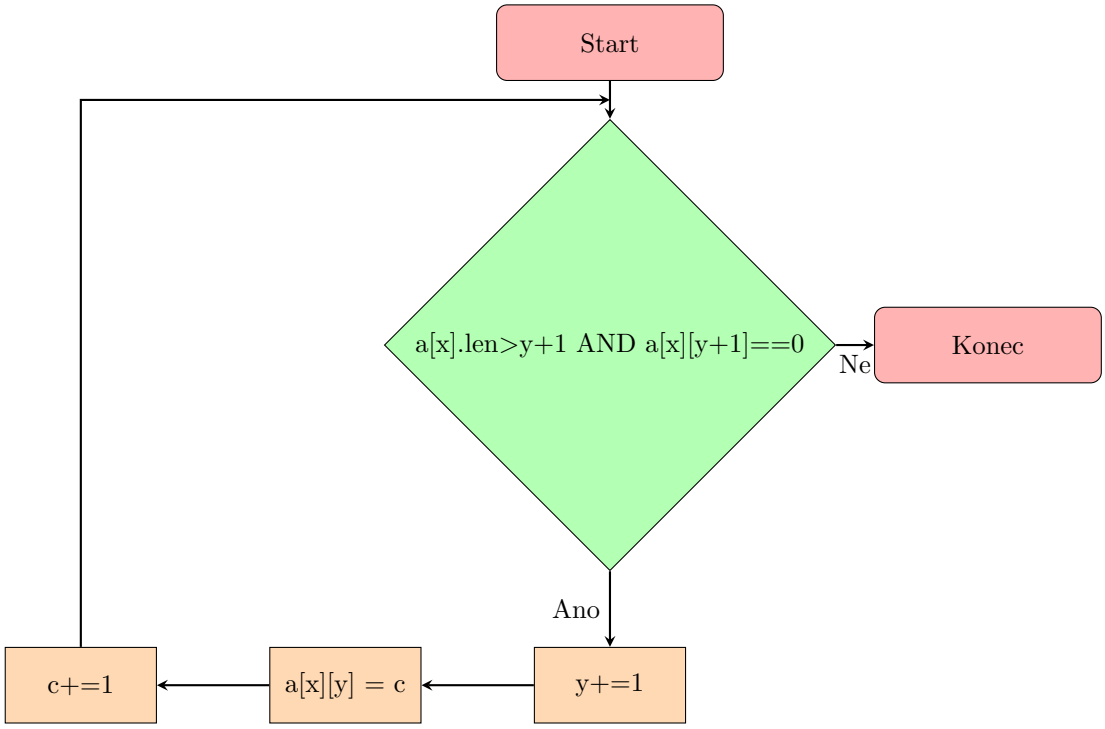
2.4 Vývojové diagramy

Následující vývojové diagramy představují jednotlivé naprogramované funkce. Pro lepší přehledenost bude pole coord nahrazeno proměnnými x, y, c.

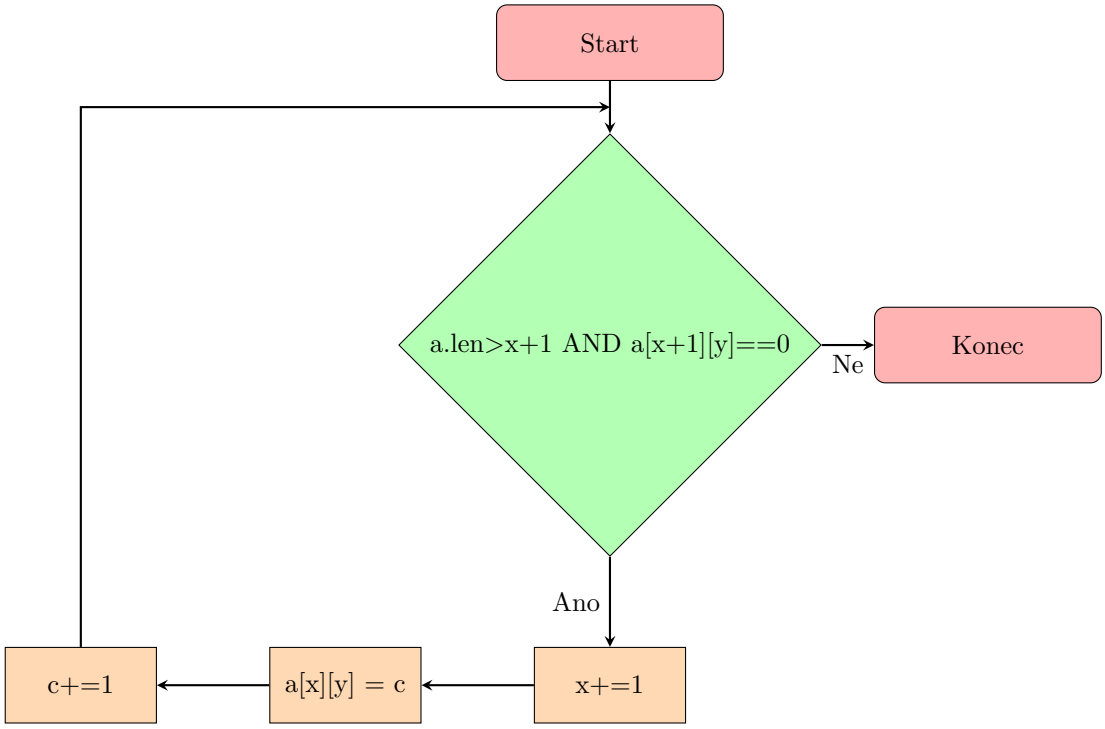
2.4.1 Metoda makeArray



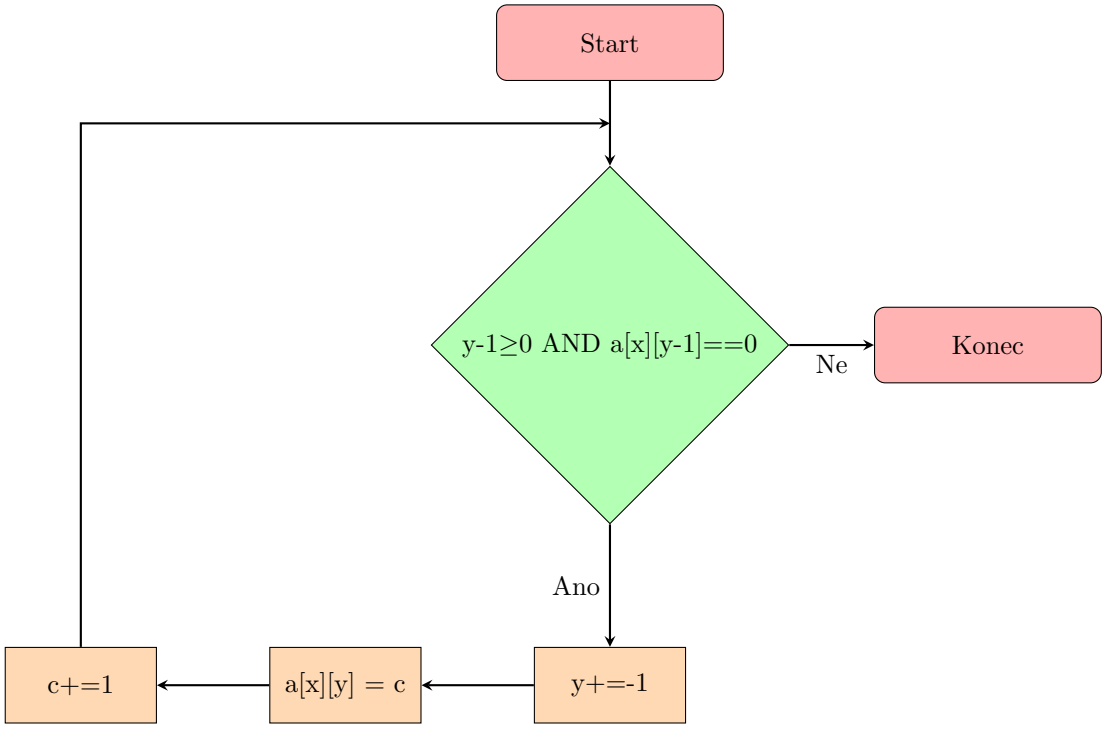
2.4.2 Metoda goRight



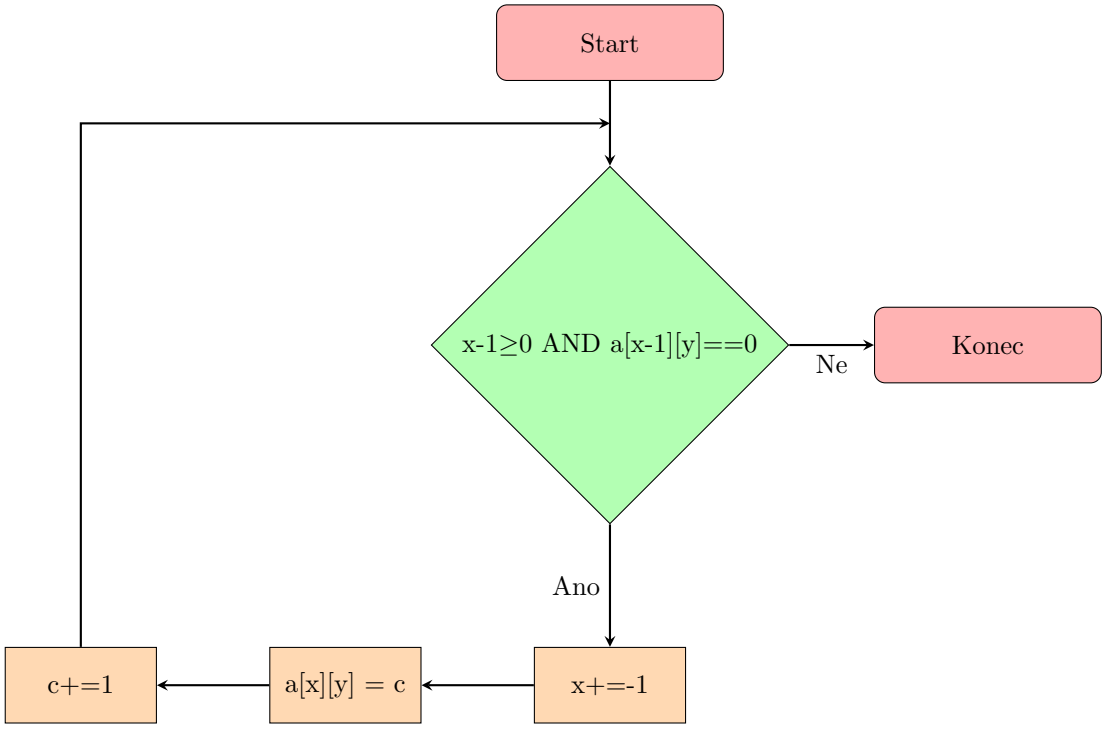
2.4.3 Metoda goDown



2.4.4 Metoda goLeft



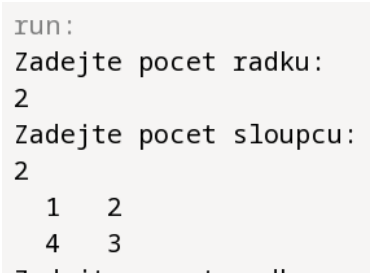
2.4.5 Metoda goUp



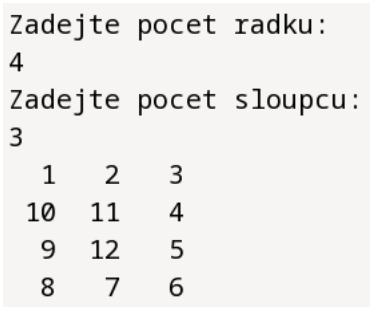
3 Protokol z testování

Číslo testu	typ testu, popis vstupů	očekávaný výsledek	Skutečný výsledek	Prošel
1	m=2;n=2	1 2 4 3	1 2 4 3	ANO
2	m=4;n=3	1 2 3 10 11 4 9 12 5 8 7 6	1 2 3 10 11 4 9 12 5 8 7 6	ANO
3	Ukonceni programu m=-1	Konec Programu	Konec Programu	ANO
4	m=1;n=1	1	1	ANO
5	m=2;n=-1	Neplatny vstup Zadejte pocet sloupcu:	Neplatny vstup Zadejte pocet sloupcu:	ANO

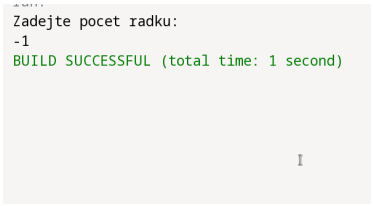
4 Screenshoty výsledků akceptačních testů



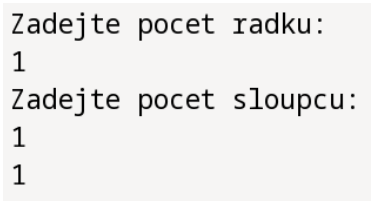
Test 1



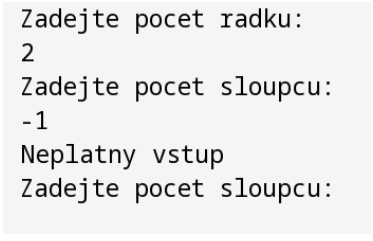
Test 2



Test 3



Test 4



Test 5