Katedra
Informatyki i Automatyki
Politechniki Rzeszowskiej

Informatyka

2.Komentarze do schematów blokowych/kodowania liczb

Opracował: Maciej Penar

Spis treści

1. Kodowanie liczb binarnych

SYSTEMY LICZBOWE:

- Addytywne np. Rzymski
- Pozycyjne np. dziesiątkowy

Pozycyjny system liczbowy składa się z podstawy p oraz zbioru znaków V, którego to elementom daje się przypisać wartości od 0 do $p-1$.

W zasadzie wartość dziesiętna liczby daje się wyznaczyć z wielomianu:
$\sum_{i=n}^{m} p^i v_i$, gdzie $p > 1, n \leq 0 \leq m$

Przykład

Jaka jest wartość dziesiętna liczby 2021,1 której podstawa to $p = 3$?
Komentarz 1: Zbiór znaków (oraz wartościowań) V:={0,1,2}
Komentarz 2: Liczbę o podstawie innej niż 10 zapisujemy $liczba_{(p)}$, w naszym przypadku 2021,1 $_{(3)}$

i	3	2	1	0	-1
v_i	2	0	2	1	1
p^i	$p^3 = 27$	$p^2 = 9$	$p^1 = 3$	$p^0 = 1$	$p^{-1} = \frac{1}{3}$
$p^i v_i$	$27 * 2 = 54$	$9 * 0 = 0$	$3 * 2 = 6$	$1 * 1 = 1$	$\frac{1}{3} * 1 = \frac{1}{3}$

$$wartość = \sum_{i=-1}^{3} p^i v_i = 54 + 0 + 6 + 1 + \frac{1}{3} = 61\frac{1}{3}$$

Przykład

Jakie jest kodowanie liczby $50_{(10)}$ gdy podstawa to $p = 3$?
Komentarz 1: Zbiór znaków (oraz wartościowań) V:={0,1,2}
Komentarz 2: Liczbę o podstawie innej niż 10 zapisujemy $liczba_{(p)}$, w naszym przypadku 2021,1 $_{(3)}$

i	3	2	1	0
p^i	$p^3 = 27$	$p^2 = 9$	$p^1 = 3$	$p^0 = 1$
$p^i v_i$	$27 * \mathbf{1} = 27$	$9 * \mathbf{2} = 18$	$3 * \mathbf{1} = 3$	$1 * \mathbf{2} = 2$
reszta	**50** – 27 = **23**	**23** – 18 = **5**	5 – 3 = 2	2 – 2 = 0
v_i	1	2	1	2

$kodowanie = 1212_{(3)}$

KODOWANIE NBC - NATURALNY KOD BINARNY

System pozycyjny o $p = 2$. Tym samym zbiorem znaków jest $V \coloneqq \{0,1\}$ (czasem też $V \coloneqq \{FALSE, TRUE\}$.

Konkretyzacja w postaci wielomianu: $\sum_{i=n}^{m} 2^i v_i$

Przykład
Jaka jest wartość dziesiętna liczby $1001{,}0101_{(2)}$

i	3	2	1	0	-1	-2	-3	-4
v_i	1	0	0	1	0	1	0	1
p^i	8	4	2	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
$p^i v_i$	8	0	0	1	0	$\frac{1}{4} = \frac{4}{16}$	0	$\frac{1}{16}$

$$wartość = \sum_{i=-4}^{3} 2^i v_i = 8 + 1 + \frac{4}{16} + \frac{1}{16} = 8\frac{5}{16}$$

Dwie uwagi:

- NBC reprezentuje tylko wartości >= 0
- Liczba o n pozycjach ułamkowych reprezentuje liczbę z dokładnością do $\frac{1}{2^{n+1}}$

KODOWANIE ZM – ZNAK MODUŁ

Ze względu na ograniczony zakres wartości kodowania NBC (tylko nieujemne wartości) – wprowadzamy specjalny bit na najbardziej znaczącej pozycji – **bit znaku.**

Wartość dziesiętna w kodowaniu ZM daje się wyznaczyć ze wzoru:
$(-1)^m \sum_{i=n}^{m-1} p^i v_i$, gdzie $p > 1, n \leq 0, m \geq 1$

$m \geq 1$ – wynika z tego że liczby w kodowaniu ZM muszą mieć co najmniej dwie pozycje (co najmniej 1 na wartość, 1 na znak liczbę)

Przykład
Jaka jest wartość dziesiętna liczby $\mathbf{0}1001_{(ZM\ 2)}$ Odpowiedź: taka jak $1001_{(2)} = 9$ Jaka jest wartość dziesiętna liczby $\mathbf{1}1001_{(ZM\ 2)}$ Odpowiedź: taka jak $-1001_{(2)} = -9$ Jaka jest wartość dziesiętna liczby $\mathbf{1}1_{(ZM\ 2)}$ Odpowiedź: taka jak $-1_{(2)} = -1$

Jaka jest wartość dziesiętna liczby $1_{(ZM\ 2)}$ Odpowiedź: nie wiadomo

Problemy:

- Dwie reprezentacje 0: 10…0 oraz 0…0
- Artymetyka

KODOWANIE U1

Wartość dziesiętna w kodowaniu U1 daje się wyznaczyć ze wzoru:
$-(p^m v_m - 1) + \sum_{i=n}^{m-1} p^i v_i$, gdzie $p > 1, n \leq 0, m \geq 1$

Najwyższy bit ma wagę: $-(p^m v_m - 1)$

$m \geq 1$ – wynika z tego że liczby w kodowaniu U1 muszą mieć co najmniej dwie pozycje (co najmniej 1 na wartość, 1 na wagę)

Przykład
Jaka jest wartość dziesiętna liczby $\mathbf{0}1001_{(U1\ 2)}$ Odpowiedź: taka jak $1001_{(2)} = 9$ Jaka jest wartość dziesiętna liczby $\mathbf{1}1001_{(U1\ 2)}$ Odpowiedź: taka jak $-1111_{(2)} + 1001_{(2)} = -15 + 9 = -6$

Komentarz:

- Dodatnie liczby liczymy jak NBC
- Ujemne – zliczamy zera

Przykład

Jaka jest wartość dziesiętna liczby $110001_{(U1\ 2)}$

i	5	4	3	2	1	0
v_i	1	1	**0**	**0**	**0**	1
p^i	-----	16	8	4	2	1
$wartości$	ujemna	0	$\mathbf{8}$	$\mathbf{4}$	**2**	0

$wartość = -(8 + 4 + 2) = -14$

Problemy:

- Dwie reprezentacje 0: 00…00 oraz 11…11
- Łatwiejsza arytmetyka, ale dalej się nie zgadza

KODOWANIE U2

Wartość dziesiętna w kodowaniu U2 daje się wyznaczyć ze wzoru:

$-p^m v_m + \sum_{i=n}^{m-1} p^i v_i$, gdzie $p > 1, n \leq 0, m \geq 1$

Najwyższy bit ma wagę: $-p^m v_m$

$m \geq 1$ – wynika z tego że liczby w kodowaniu U2 muszą mieć co najmniej dwie pozycje (co najmniej 1 na wartość, 1 na wagę)

Przykład

Jaka jest wartość dziesiętna liczby $110001_{(U1\ 2)}$

i	5	4	3	2	1	0
v_i	1	1	0	0	0	1
p^i	32	16	8	4	2	1
$p^i v_i$	32, ale ujemna -32	16	0	0	0	1

$wartość = 32 - (16 + 1) = -15$

Częstsza forma konwersji to inwersja + 1.

Przykład

Jaka jest zakodować 19 w U2?
NBC: 10011
U2: inv(NBC) + 1 = inv(10011) + 1 = 01100 + 1 = 01101 = (dodajemy bit wagi na początek) = $101101_{(U1\ 2)}$

Cechy:

- Arytmetyka działa prawidłowo
- Jedna reprezentacja zera

KODOWANIE Z NADMIAREM - BIAS[X]

Koncepcja: przesuwamy zakres wartości o stałą x. Liczy się tak samo jak w przypadku NBC z dodatkowym czynnikiem:

W zasadzie wartość dziesiętna liczby daje się wyznaczyć z wielomianu:
$(\sum_{i=n}^{m} p^i v_i) - x$, gdzie $p > 1, n \leq 0 \leq m$

Przykład

Jaka jest wartość dziesiętna liczby $010101_{(BIAS\ 16)}$

i	4	3	2	1	0
v_i	1	0	1	0	1
p^i	16	8	4	2	1
$p^i v_i$	16	0	4	0	1

$wartość = 16 + 4 + 1 - 16\ (bias) = 5$

Przykład

Mam wartość 23, jak wygląda BIAS[16]

Najpierw znajdę reprezentację NBC:

i	4	3	2	1	0
$\boldsymbol{v_i}$	**1**	**0**	**1**	**1**	**1**
p^i	16	8	4	2	1
$p^i v_i$	16	0	4	2	1

$wartość_{NBC} = 16 + 4 + 2 + 1 = 23$, czyli reprezentacja NBC to 10111
BIAS[16] = NBC + x = 1 0111 + 1 0000 = $\mathbf{10\ 0111}_{(BIAS\ 16)}$

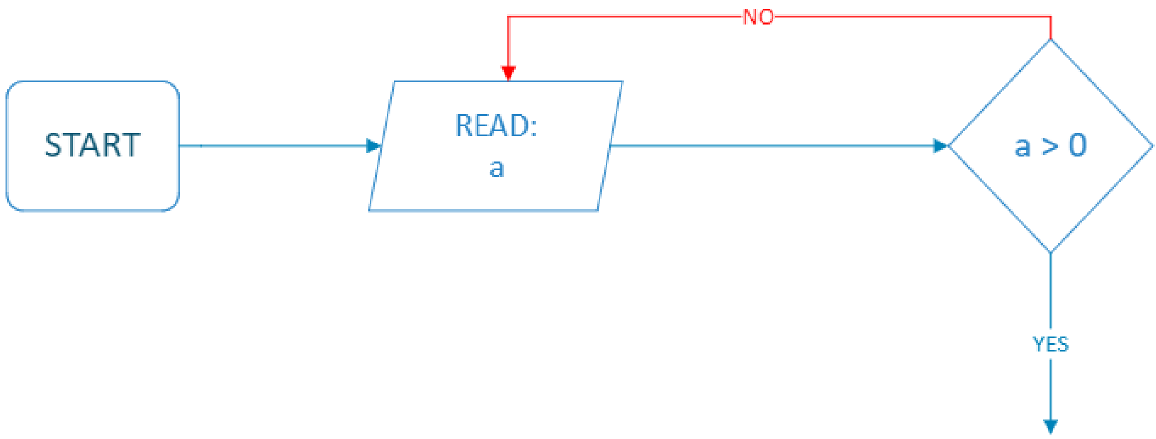
Cechy:

- Arytmetyka nie działa
- Jedna reprezentacja zera
- Rozszerzanie liczby (dodawanie pozycji) zmienia tylko zakres dodatnich wartości – co wprowadza asymetrię

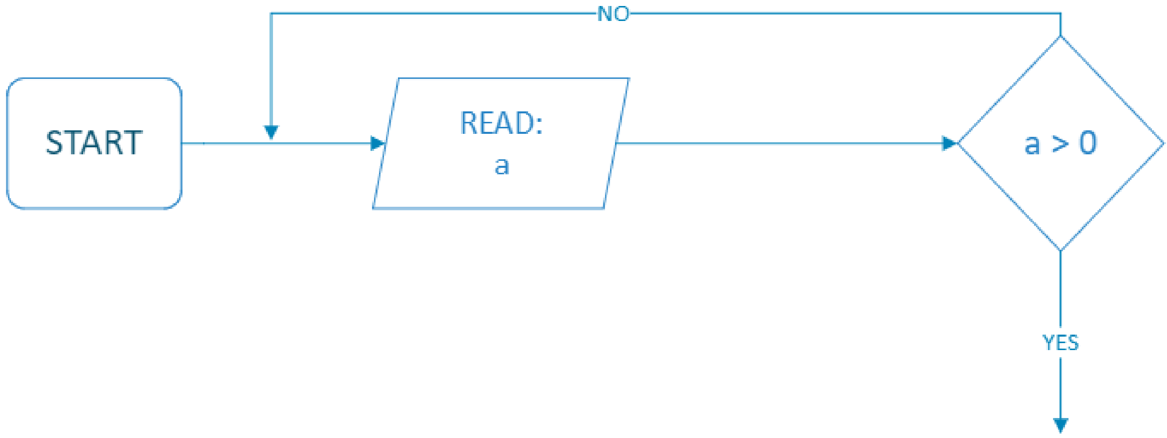
2. Schematy blokowe

SCIENCE FICTION DOT. SCHEMATÓW BLOKOWYCH

Czasem pojawia się chęć błędnego zapisu:



Zamiast poprawnego:

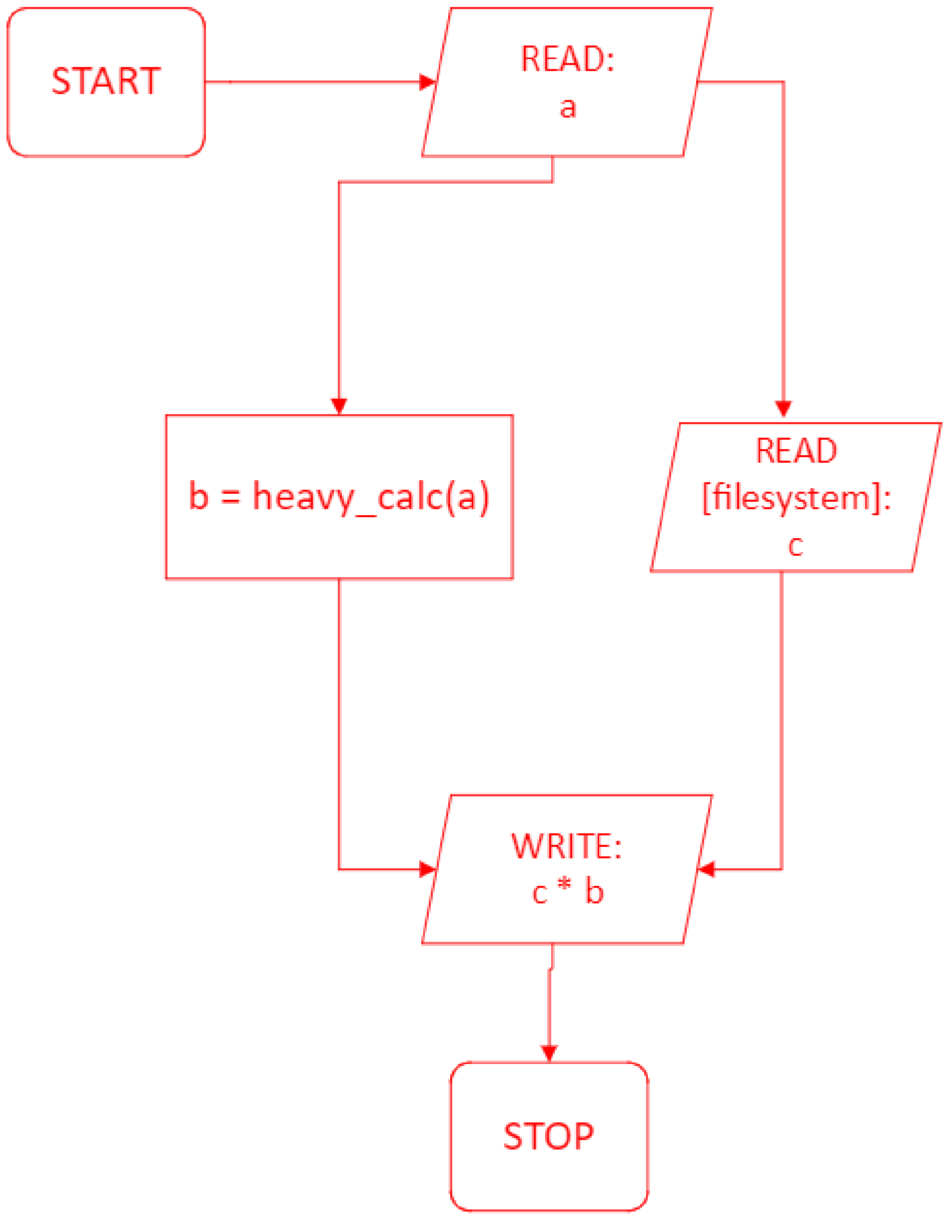


Można pomyśleć – jaka jest różnica? Otóż można pomyśleć o zapisie błędnym jako o wyrażaniu synchronizacji dwóch różnych potoków przetwarzania (podobny semantyka obowiązuje przy diagramach aktywności / BPMN / sieciach Petriego).

Bardzo szybko diagramy blokowe algorytmów zostaną zastąpione przeze pseudokod/kod więc nie będzie okazji do wyrażania algorytmów w których obliczenia wykonywane są równolegle. Nie ma takiej potrzeby. Nigdy nie było.

Niemniej jednak można czysto konceptualnie zastanowić się co to oznacza.

Nie myśleć o tym za dużo i NIE przyzwyczajać się

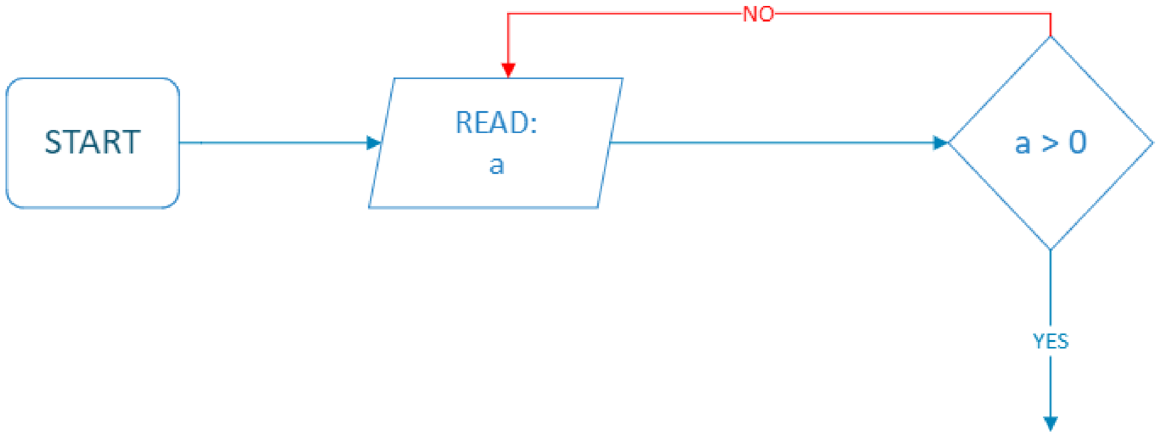


Komentarz:
Taki diagram mógłby wyrażać:

1) Po wczytaniu a – następuje fork na dwa osobne potoki
   a. Wyliczający b (proces powiedzmy trwa 5 sekund w RAM)
   b. Wczytujący dane z systemu plików (trwa 3 sekundy z HDD)
2) Zwrócenie c*b ma tylko sens gdy oba potoki się zakończyły, więc podwójne wejście oznacza **oczekiwanie na koniec obliczeń wszystkich bloków wejściowych**
3)

Nikt nie zaprząta sobie głowy wyrażaniem współbieżności na schematach blokowych algorytmów. Tym samym jest to **błąd**, ponieważ wprowadza niejednoznaczność do diagramu.

Wracając do schematu blokowego:

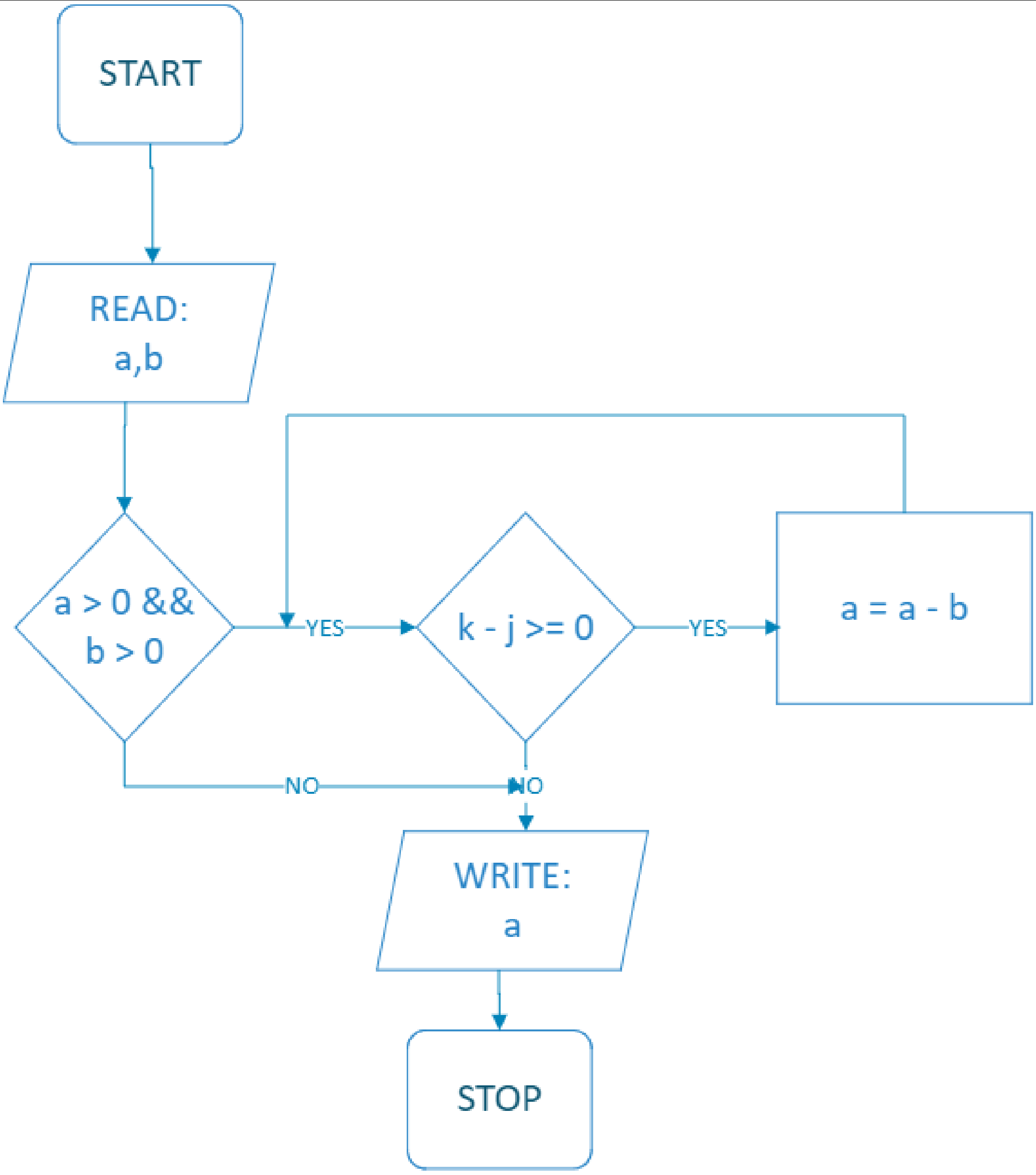


W takim rozumieniu współbieżności oznaczałoby blokadę (deadlock) ponieważ blok 'READ a' wykona się po START oraz ewaluacji warunku a>0 – to nie jest możliwe.

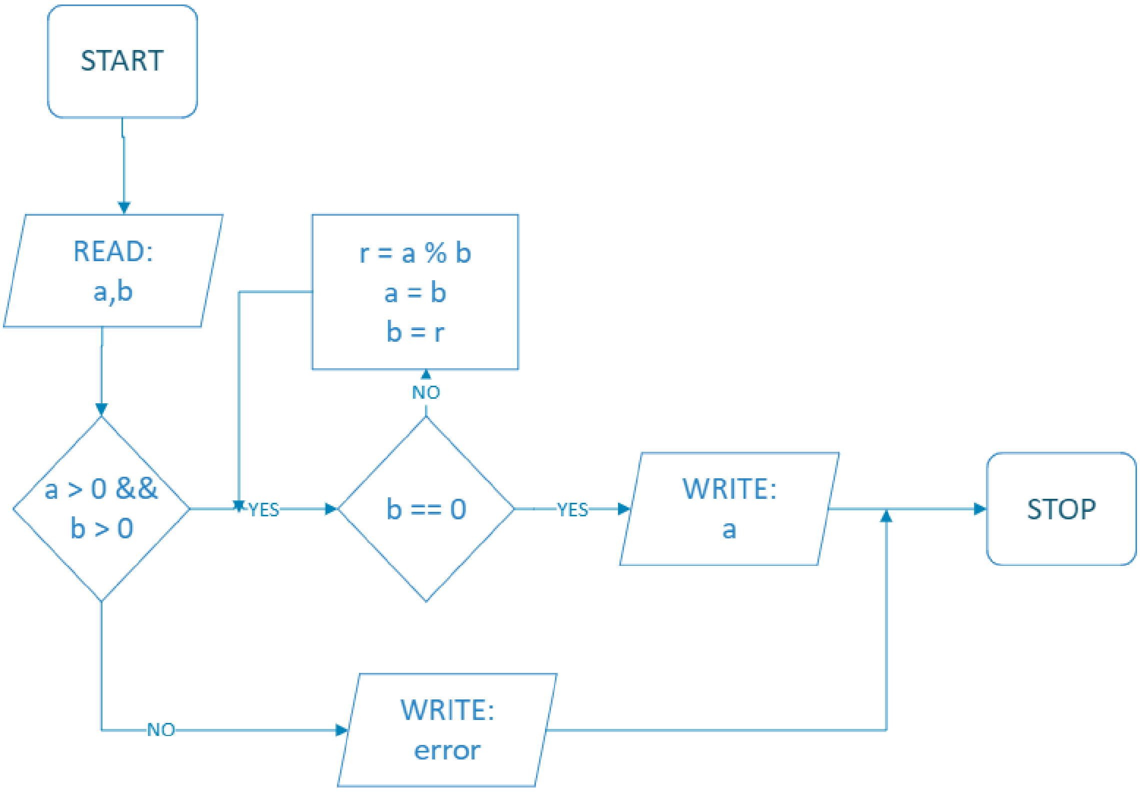
ZADANIA KTÓRE BYŁY NA ZAJĘCIACH:

1. Wczytać a, b. Zwrócić resztę z dzielenia $(a\ mod\ b)$ – ale bez operacji mod
2. Wczytać a, b. Zwrócić Największy Wspólny Dzielnik a, b.
3. Wczytać a, b. Zwrócić Najmniejszą Wspólną Wielokrotność a, b. [Policzyliśmy już NWD(a,b), wzór na NWW(a,b) = $NWW(a,b) = \frac{ab}{NWD(a,b)}$.
4. Wczytać liczbę współczynników wielomianu (n) oraz współczynniki do tablicy oraz wartość x. Zwrócić $\sum_{i=0}^{n-1} x^i a_i$.
5. Wczytać tablicę. Posortować przez jakąkolwiek (naiwną) implementację Sortowania Bąbelkowego. Zwrócić tablicę.
6. Wczytać zakres liczb (m) oraz tablicę liczb. Posortować przez sortowanie przez zliczanie. LINK: https://pl.wikipedia.org/wiki/Sortowanie_przez_zliczanie
7. Wczytać liczbę n. Znaleźć rozkład na czynniki pierwsze. Wypisać.
8. Wczytać rok, miesiąc, dzień. Wykonać walidację daty. Zwrócić „Poprawny" „Niepoprawny".

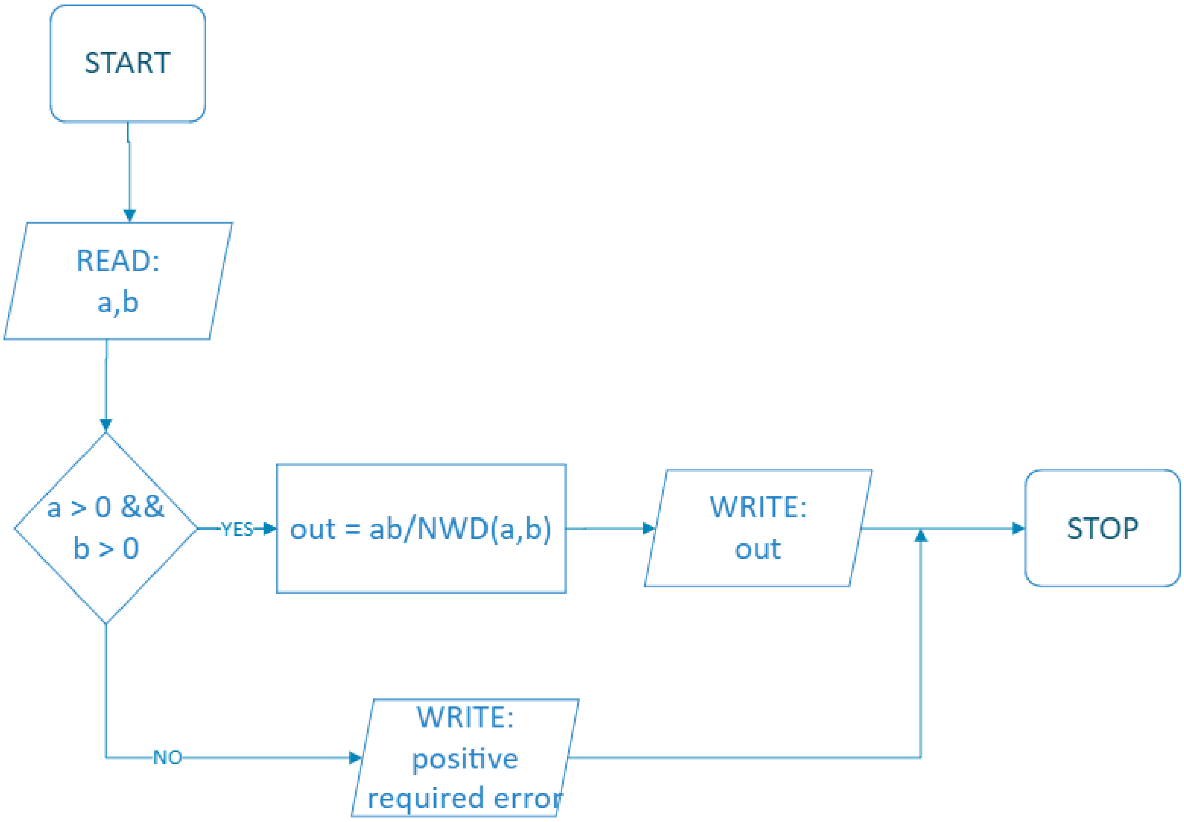
ROZWIĄZANIA

1.Wczytać a, b. Zwrócić resztę z dzielenia $(a \ mod \ b)$ – ale bez operacji mod


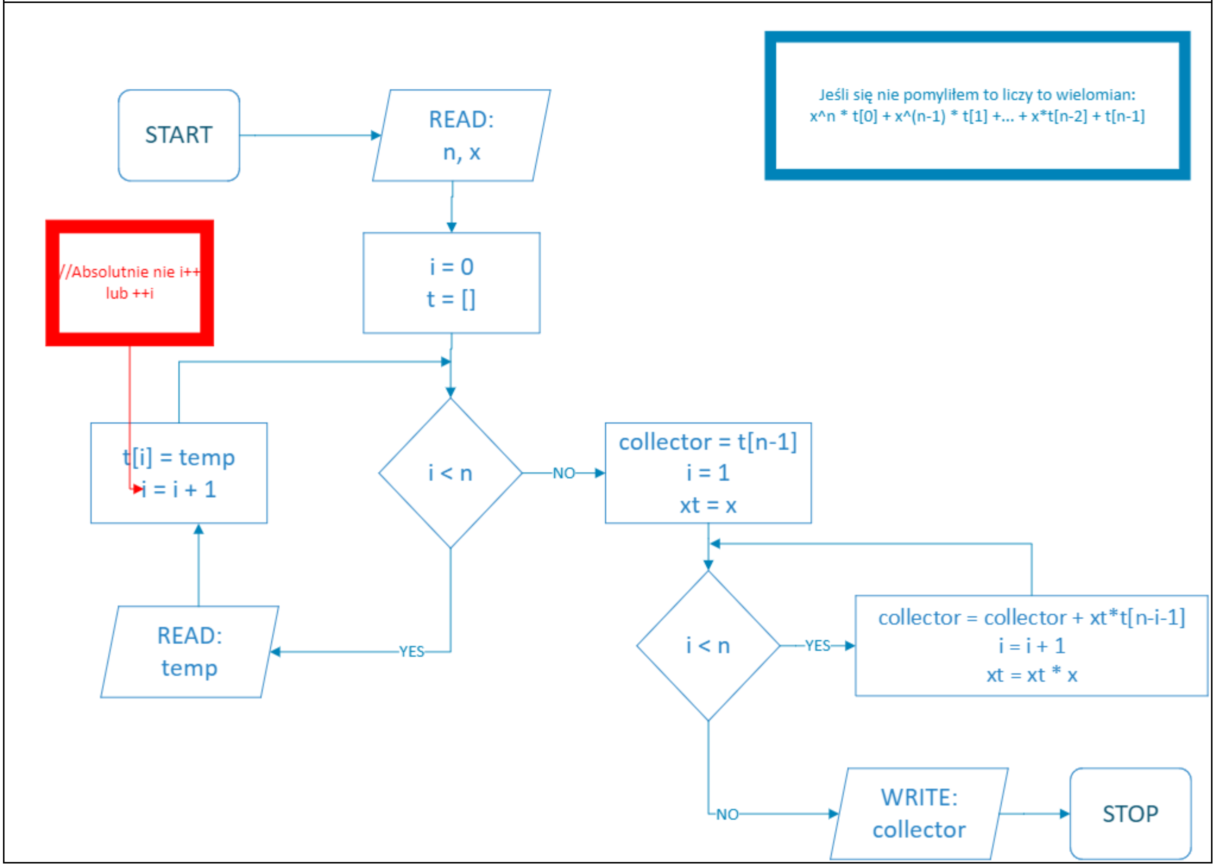
2. Wczytać a, b. Zwrócić Największy Wspólny Dzielnik a, b.



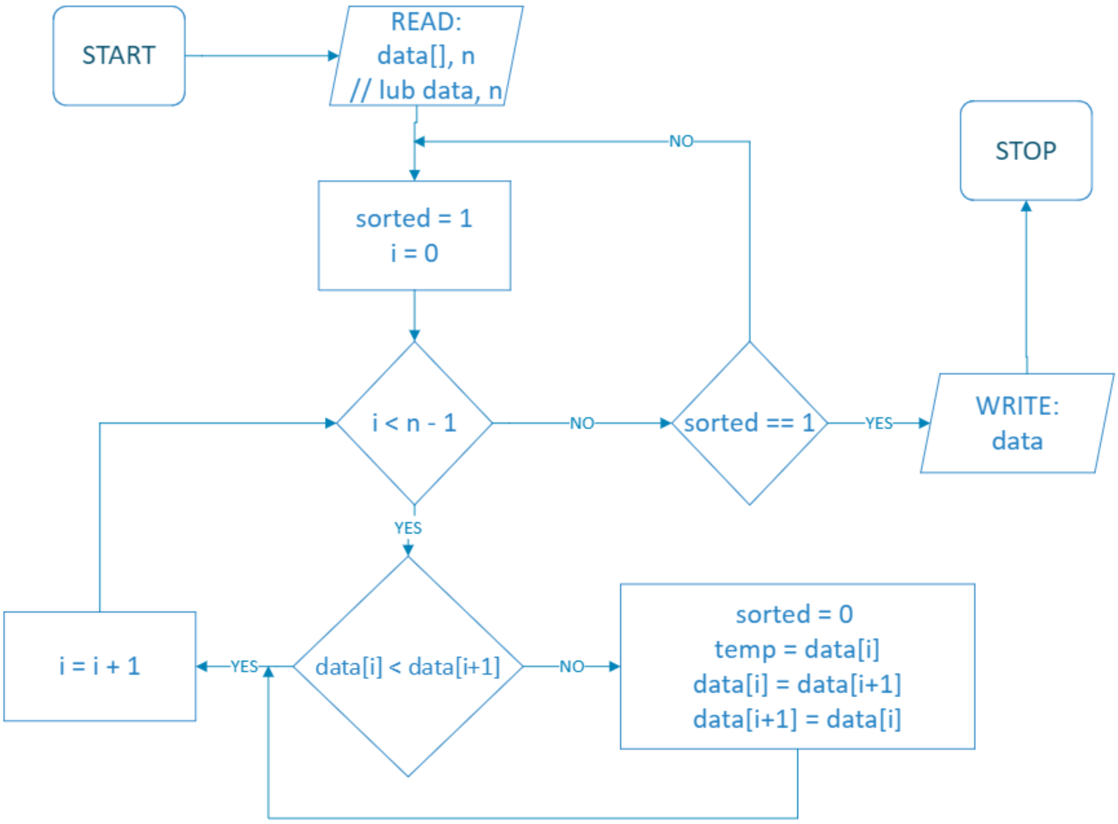
3. Wczytać a, b. Zwrócić Najmniejszą Wspólną Wielokrotność a, b



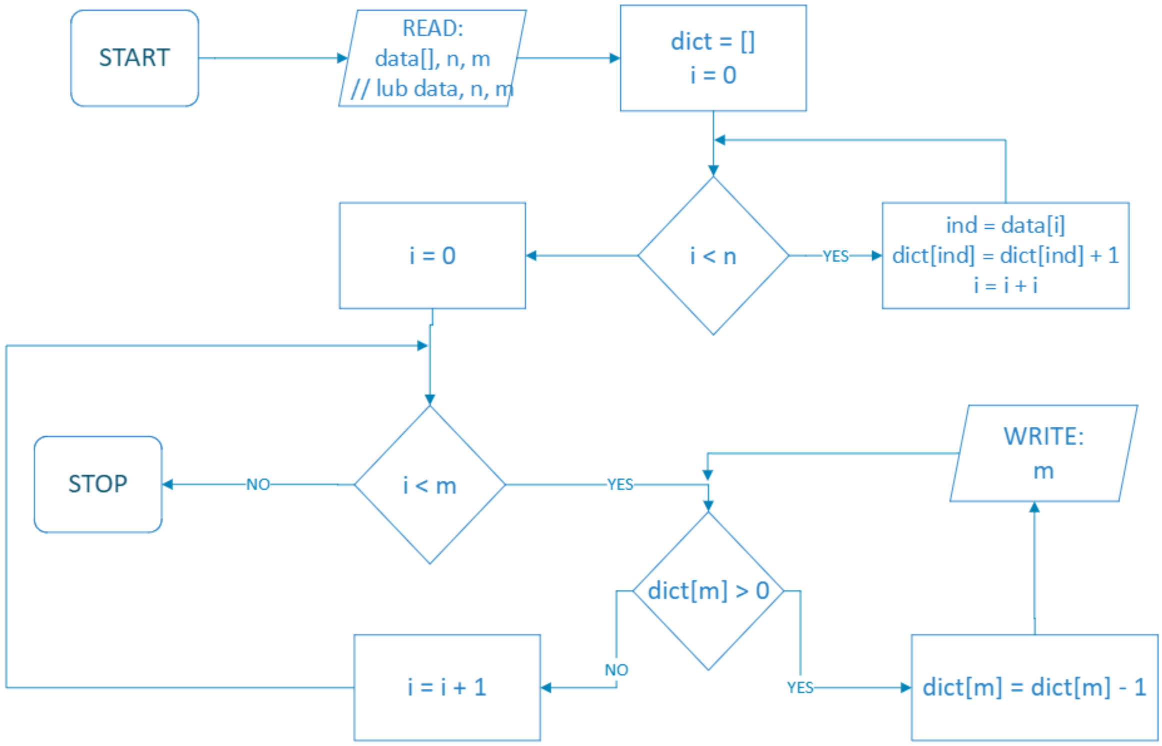
4. Wczytać liczbę współczynników wielomianu (n) oraz współczynniki do tablicy oraz wartość x



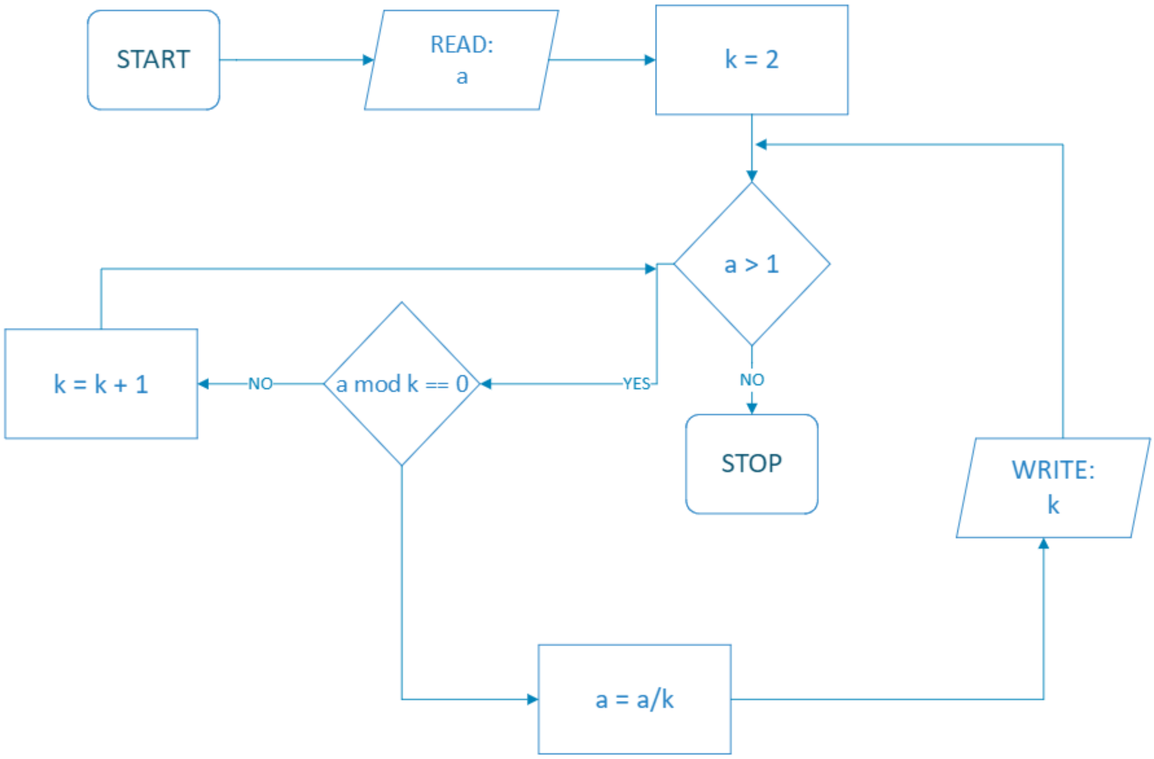
5. Wczytać tablicę. Posortować przez jakąkolwiek (naiwną) implementację Sortowania Bąbelkowego.



6. Wczytać zakres liczb (m) oraz tablicę liczb. Posortować przez sortowanie przez zliczanie



7. Wczytać liczbę n. Znaleźć rozkład na czynniki pierwsze.



8. Wczytać rok, miesiąc, dzień. Wykonać walidację daty. Zwrócić „Poprawny” „Niepoprawny”.
LINK: https://www.youtube.com/watch?v=dQw4w9WgXcQ