Zadania dodatkowe 1

1. **Obliczanie ceny z podatkiem**

   Paragon fiskalny zawiera następujące informacje: cenę jednostkową netto *a*, liczbę zakupionych sztuk towaru *x* oraz wysokość podatku VAT podaną w procentach *p*. Napisz program, który wczytuje powyższe dane z klawiatury, a następnie oblicza wartość netto zakupionego towaru, wartość podatku VAT oraz wartość towaru brutto. Wyniki obliczeń należy wypisać na ekranie w formie uproszczonego „paragonu fiskalnego”.
   Wskazówki:
   - Wartość netto: $wn = a * x$
   - Podatek VAT: $v = w * p /100$
   - Wartość butto: $wb = wn + v$

2. **Obliczanie wartości wielomianu za pomocą wzoru Hornera**

   Wielomian trzeciego stopnia zapisany jest wzorem: $W(x) = a\ x^3 + b\ x^2 + c\ x + d$ Napisz program, który oblicza wartość tego wielomianu za pomocą wzoru Kornera:
   $W(x) = ((a\ x + b)\ x + c)\ x + d.$
   Wskazówki:
   - wartości parametrów *a*, *b*, *c*, *d* oraz wartość zmiennej *x* należy odczytać z klawiatury,
   - wartość wielomianu obliczyć następująco:
     *w= a*x+b*
     *w= w*x+c*
     *w= w*x+d*
   - końcową wartość wielomianu należy wyświetlić na ekranie.

3. **Rozwiązywanie równania kwadratowego**

   Napisz program, który oblicza pierwiastki równania kwadratowego: $a\ x^2 + b\ x + c = 0.$ Program powinien wczytywać współczynniki *a*, *b*, *c* z klawiatury, następnie sprawdzać czy jest to poprawne równanie kwadratowe (tzn. czy *a* jest różne od *0*), obliczać wartość wyróżnika *delta* oraz jeśli istnieją to obliczać pierwiastki *x1* i *x2*. Wyniki należy wypisać na ekranie.
   Wskazówki:
   - Współczynniki *a*, *b*, *c* należy odczytać z klawiatury,
   - jeśli $a = 0$ to brak równania kwadratowego – nie można liczyć pierwiastków,
   - *delta = b2 - 4*a*c,*
   - jeśli *delta < 0* to brak pierwiastków rzeczywistych,
     jeśli *delta = 0* to istnieje podwójny pierwiastek *x1 = x2 = -b/(2*a)*
     jeśli *delta > 0* to *x1 = (-b – sort(delta))/(2*a)*
     *x2 = (-b + sort(delta))/(2*a)*

4. **Obliczanie sumy szeregu harmonicznego**

   Szeregiem harmonicznym pierwszego rzędu nazywamy następujący szereg liczbowy:
   $1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots$
   Napisz program, który oblicza z zadaną dokładnością $0 < eps < 1$ sumę szeregu harmonicznego. Dokładność *eps* należy wczytać z klawiatury.
   Wskazówki:

- Jeśli podana dokładność *eps <= 0* albo *eps >= 1* to należy wyświetlić komunikat o błędzie,
- suma szeregu jest równa:

  suma = 1 + 1/2 + 1/3 + 1/4 + … + 1/n, gdzie *1/n >= eps*,
- sumowanie należy przerwać gdy kolejny obliczony wyraz szeregu jest mniejszy niż *eps*.

5. Obliczanie *nwd* oraz *nww*

Napisz program, który wczytuje z klawiatury dwie liczby całkowite *a* i *b*, Dla podanych liczb należy obliczyć i wypisać na ekranie największy wspólny dzielnik *nwd* oraz najmniejszą wspólną wielokrotność *nww*.

Wskazówki:

- *nwd* jest to największa liczba całkowita, dla której:

  a % nwd = 0 oraz *b % nwd=0*,

  gdzie *x % y* jest resztą z dzielenia *x* przez *y*.
- *nww* jest to najmniejsza liczba całkowita, dla której:

  nww % a = 0 oraz *nww % b = 0*.
- Np. dla *a=6* oraz *b=9*

  największy wspólny dzielnik *nwd* wynosi *3*,

  najmniejsza wspólna wielokrotność *nww* wynosi *18*.

6. **Obliczanie wartości stałych matematycznych *pi* oraz *e***

Napisz program, który wczytuje z klawiatury pożądaną dokładność obliczeń *eps* (*eps* < *0.1*), a następnie oblicza z zadaną dokładnością wartość sumy następujących szeregów:

$$S1 = 4 * \sum_{k=1}^{\infty} \left[(-1)^{k+1} * \frac{1}{2k-1} \right] = 4 * \left[\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots \right]$$

$$S2 = \sum_{k=0}^{\infty} \frac{1}{k!}$$

Wskazówki:

- Sumowanie szeregu należy zakończyć, gdy wartość bezwzględna pojedynczego wyrazu szeregu jest mniejsza niż zadana dokładność obliczeń *eps*.
- Suma szeregu *S1* jest zbieżna do wartości stałej *pi*
- suma szeregu *s2* jest zbieżna do wartości stałej *e*.

7. Obliczanie funkcji $\pi(n)$

Napisz funkcję, która wczytuje z klawiatury dodatnią liczbę całkowitą *n*, a następnie dla podanej liczby *n* oblicza wartość funkcji $\pi(n)$ czyli ilość liczb pierwszych mniejszych lub równych *n*. Dla *n>3* postać funkcji $\pi(n)$ jest następująca:

$$\pi(n) = -1 + \sum_{j=3}^{n} \left((j-2)! - j \left\lfloor \frac{(j-2)!}{j} \right\rfloor \right)$$

gdzie $\lfloor x \rfloor$ oznacza część całkowitą *x* zwracaną przez funkcje `floor(x)`, natomiast `x!` oznacza silnię liczby `x`.

Wskazówki:

- Funkcja `floor(x)` w języku Java jest obliczana za pomocą metody `Math.floor(x).`

8. Obliczanie ciągu Hofstadtera

Napisz funkcję, która wczytuje z klawiatury dodatnią liczbę całkowitą n_0 , a następnie generuje ciąg liczb całkowitych zgodnie z następującym wzorem:

$$n_{i+1} = \begin{cases} n_i / 2 \quad jesli \quad n_i \quad jest \ parzyste, \\ 3 * n_i + 1 \quad jesli \quad n_i \quad jest \ nieparzyste. \end{cases}$$

Obliczenia należy zakończyć, gdy kolejna obliczona liczba n_{i+1} będzie równa 1. Funkcja powinna wyświetlać na ekranie w kolejnych wierszach następujące informacje: numer iteracji i+1, wartość n_i, {parzyste|nieparzyste}, wartość n_{i+1}

np. dla $n_0 = 10$ na ekranie powinno pojawić się:

```
1, 10, parzyste, 5
2, 5, nieparzyste, 16
3, 16, parzyste, 8
4, 8, parzyste, 4
5, 4, parzyste, 2
6, 2, parzyste, 1
```

Wskazówki:

- Algorytm ten zaproponował Hofstadter. Co ciekawe, nie ma dowodu, że algorytm zawsze się zatrzymuje. Jaki jest ciąg wynikowy dla liczby 27 ?