2022-03-31_Zadanie domowe

Z1.
Napisz w języku Python program, który w zależności od decyzji użytkownika sprawdza, czy dana liczba trzycyfrowa jest liczbą Armstronga albo wypisuje wszystkie liczby Armstronga zawierające trzy cyfry (wersja trudniejsza). Liczba Armstronga, zwana również liczbą narcystyczną, to liczba, która jest sumą swoich cyfr podniesionych do potęgi równej ich liczbie. Przykładowo, najmniejsza liczba Armstronga to 153, która jest równa $1^3 + 5^3 + 3^3$.

Z2.
Napisz w języku Python program, który w zależności od decyzji użytkownika sprawdza, czy dana liczba jest obfita albo wyświetli liczby obfite z zakresu podanego przez użytkownika oraz obliczy ich obfitość. Liczba obfita jest liczbą, dla której suma jej dzielników właściwych jest większa od niej samej. Wartość, o jaką suma dzielników przekracza liczbę, nazywamy obfitością. Na przykład liczba 12 jest liczbą obfitą, bo suma jej dzielników właściwych wynosi $1 + 2 + 3 + 4 + 6 = 16$, a jej obfitość wynosi $16 - 12 = 4$.

Z3.
Liczbę naturalną nazywamy osiągalną, jeżeli istnieje takie k, że $n = k + s(k)$, gdzie k jest liczbą naturalną, a $s(k)$ jest sumą cyfr liczby k w zapisie dziesiętnym. Przykładowo, liczba $n = 28$ jest osiągalna, ponieważ istnieje $k = 23$, w której $s(k) = 5$, zaś $23 + 5 = 28$. Podobnie osiągalną liczbą jest $n = 505$, bo istnieje takie $k = 491$ oraz $s(k) = 14$, że $491 + 14 = 505$. Natomiast np. liczby 20 i 31 nie są osiągalne. Na podstawie powyższych informacji napisz program w języku Python, który w zależności od decyzji użytkownika sprawdza, czy podana przez użytkownika liczba z przedziału $[10;\ 9999]$ jest osiągalna albo generuje wszystkie liczby osiągalne z tego zakresu .

Z4.
Kod Graya, znany również pod nazwą kodu refleksyjnego lub odzwierciedlonego binarnie, jest formą kodowania binarnego, w którym dwie kolejne liczby różnią się od siebie tylko jednym bitem.

Decimal	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Napisz w języku Python program, który przekształca kod binarny do kodu Graya i odwrotnie.

Z5.

Dana jest prostokątna tablica T_{mn}, gdzie m oznacza liczbę wierszy, a n liczbę kolumn. Tablica wypełniona jest przez użytkownika wyłącznie zerami i jedynkami. Dla przykładu, użytkownik utworzył tablicę T_{56} i wypełnił ją zerami i jedynkami w następujący sposób:

1	1	1	0	1	0
0	**1**	1	1	0	1
1	**1**	0	0	0	1
0	**1**	**1**	**1**	0	1
1	0	0	**1**	0	1

Ciąg jedynek przyległych w poziomie lub w pionie tworzy drogę. Należy sprawdzić, czy między wskazanym przez użytkownika punktem początkowym i końcowym *(tutaj: punkt początkowy o współrzędnych 2,2 i końcowy o współrzędnych 5,4 zaznaczono dla ułatwienia podkreśleniem)* istnieje droga. W powyższej tablicy droga taka istnieje i została oznaczona pogrubioną czcionką.

Zadanie polega na opracowaniu algorytmu i zaimplementowaniu go w języku Python.

Z6.

Silniowy system pozycyjny to pozycyjny zapis liczb naturalnych, w którym mnożniki dla kolejnych pozycji definiowane są przez silnie kolejnych liczb naturalnych

$$(x)_! = (x_n x_{n-1} x_{n-2} \cdots x_2 x_1) = x_n \cdot n! + x_{n-1} \cdot (n-1)! + \cdots + x_2 \cdot 2! + x_1 \cdot 1!$$

W systemie silniowym współczynnik x_i, który odpowiada mnożnikowi $i!$, spełnia zależność $0 \leq x_i \leq i$.

Zapis każdej liczby w silniowym systemie pozycyjnym jest jednoznaczny, tj. każdą liczbę naturalną można zapisać dokładnie i tylko w jeden sposób.

Przykład zamiany liczby w zapisie silniowym na liczbę w zapisie dziesiętnym

$$(1220)_! = 1 \cdot 4! + 2 \cdot 3! + 2 \cdot 2! + 0 \cdot 1! = (40)_{10}$$

Zamiana zapisu liczby x w systemie dziesiętnym na zapis liczby w systemie silniowym może przebiegać wg następującego algorytmu: szukamy największej liczby k, której silnia nie przekracza x. Pierwsza jej cyfra to wynik dzielenia całkowitego x przez $k!$. Kolejne cyfry zapisu silniowego (zaczynając od cyfr najbardziej znaczących) otrzymujemy przez wyznaczanie wyników dzielenia liczby x przez $(k-1)_!$, $(k-2)_!$,$\cdots$, $2!$, $1!$. Po wyznaczeniu cyfry x_i, odpowiadającej współczynnikowi $i!$, zmniejszamy wartość x o liczbę odpowiadającą cyfrze x_i, czyli $x_i \cdot i$. Oznacza to, że x przyjmuje wartość $x \; mod \; k!$.

Przykład zamiany liczby dziesiętnej 1548 na zapis silniowy

x	k	$x\ div\ k!$	$x\ mod\ k!$
1548	6	**2**	108
108	5	**0**	108
108	4	**4**	12
12	3	**2**	0
0	2	**0**	0
0	1	**0**	0

Zatem $(1548)_{10} = (204200)_!$.

Na podstawie powyższych rozważań napisz w języku Python program, który w zależności od decyji użytkownika zamienia liczbę w zapisie silniowym na liczbę dziesiętną albo liczbę dziesiętną na odpowiadającą jej liczbę w systemie silniowym.

Termin: 20 kwietnia 2022 r. (do godz. 24:00)