1.Proszę napisać algorytm: dane są dwa zbiory A i B (n-elementowy i m-elementowy) o elementach będących liczbami naturalnymi. Elementy tych zbiorów dane są za pomocą dwóch tablic **uporządkowanych rosnąco**. Proszę wyznaczyć liczbę elementów o wartościach nieparzystych należących do zbioru A ⊕ B (suma rozłączna zbiorów). Wskazana złożoność liniowa.
Np. A={1,2,3,5,8}, B={0,1,3,4,8,9,10}, A⊕ B ={0,2,4,5,9,10}, ale wartości nieparzyste to: 5,9. Liczba szukanych elementów to 2.

2. Dane są dwie tablice liczb całkowitych A i B o elementach **uporządkowanych niemalejąco**. Proszę wyznaczyć liczbę jednokrotnych elementów należących zarówno do tablicy A i B.
Np. A={1,2,2,2,3,4,4,4,5,8,10}, B={2,2,4,4,4,4,10,16,20}. Wspólne elementy to {2,2,4,4,4,10} ale tylko trzy występują jednokrotnie. Są to{2,4,10}, więc szukana liczba to 3. Wskazana złożoność liniowa.

3. Proszę napisać algorytm: dane są dwa zbiory A i B (n-elementowy i m-elementowy) o elementach będących liczbami naturalnymi oraz zadana wartość (liczna naturalna). Elementy tych zbiorów dane są za pomocą dwóch **tablic uporządkowanych rosnąco**. Proszę wyznacz liczbę elementów należących do zbioru A \ B (różnica zbiorów A-B) i większych od zadanej wartości. Zalecana złożoność liniowa.
Np. A={1,2,3,5,8}, B={0,1,3,4,8,9,10} i zadana wartość to 3. A\B={2,5} ale istnieje jeden element większy od 3. Jest to wartość 5.

4.Proszę napisać algorytm: dane są dwie tablice A i B (n-elementowa i m-elementowa) o elementach będących liczbami naturalnymi. Tablice są **uporządkowanych niemalejąco**. Proszę wyznaczyć liczbę elementów należących do A ∪ B (suma zbiorów tzn. każdy element może wystąpić Jednokrotnie). Wskazana złożoność liniowa.
Np. A={1,2,2,3,3,5,8}, B={1,3,4,8,9,10}, A ∪ B ={1,2,3,4,5,8,9,10}, więc szukana liczba to 8.

5. Dane są dwie tablice liczb całkowitych A i B o elementach **uporządkowanych niemalejąco**. Rozmiar tablicy A >= rozmiaru tablicy B. Sprawdzić ile czy tablica B zawiera się w całości w tablicy A (bez naruszania kolejności elementów tablicy B). Wskazana złożoność liniowa.
Np. dla A={1,2,$\boxed{2,2,3,4,4,4}$,5,8,10}, B={2,2,3,4,4,} własność jest spełniona.

6. Proszę napisać algorytm: dane są dwa zbiory A i B (n-elementowy i m-elementowy) o elementach będących liczbami naturalnymi oraz zadana wartość (liczna naturalna). Elementy tych zbiorów dane są za pomocą dwóch **tablic uporządkowanych rosnąco**. Proszę wyznacz liczbę elementów należących do zbioru A \ B (różnica zbiorów A-B) i mniejszych od zadanej wartości. Zalecana złożoność liniowa.
Np. A={1,2,3,5,8}, B={0,1,3,4,8,9,10} i zadana wartość to 3. A\B={2,5} ale istnieje jeden element mniejszy od 3. Jest to wartość 2.

7. Proszę napisać algorytm, który dla dowolnej tablicy jednowymiarowej o elementach będących liczbami całkowitymi wyznacza rozmiar najdłuższej, ciągłej podtablicy niemalejącej(tzn. że każdy element tej podtablicy jest większy lub równy od poprzedniego).
Np. dla tablicy { 1 , 8 , 1 , 1 , 5 , 7 , 1 , 1 } taka podtablica jest { 1, 1 , 5 , 7 } i jej długość wynosi 4.

8. Proszę napisać algorytm, który dla dowolnej tablicy jednowymiarowej o elementach będących liczbami całkowitymi, wyznacza rozmiar najdłuższej o liczności parzystej, ciągłej podtablicy stałej(tzn. zawierającej identyczne wartości w kolejnych elementach).

Np. dla tablicy { 1 , 8 , 1 , 1 , 5 , 4 ,4 , 4 } taka podtablica jest { 1 , 1 } i jej długość wynosi 2. Jest również podtablica stała { 4, 4, 4} o większym rozmiarze, ale jej liczność wynosi 3 i jest liczbą nieparzysta.

9.Dla tablicy o wymiarze *n* i elementach będących liczbami całkowitymi, proszę napisać algorytm:

9.1 - wyznaczającą liczbę wystąpień minimalnego elementu w tablicy.

9.2 - wyznaczającą maksymalną odległość między elementami o minimalnej wartości(*uwaga: minimalna wartość w tablicy jest jedna, ale może wystąpić kilkakrotnie*).

Np. dla tablicy {1, 4, 1, 1, 7, 2, 3} minimalna wartość wynosi 1 i występuje 3 razy. Maksymalna odległość miedzy nimi wynosi 3 (różnica między indeksami ostatniego i pierwszego wystąpienia wartości 1). Jeśli wartość minimalna występuje jednokrotnie to maksymalna odległość wynosi 0.

10.Dla tablicy o wymiarze *n* i elementach będących liczbami całkowitymi, proszę napisać algorytm:

10.1 - wyznaczającą liczbę wystąpień maksymalnego elementu w tablicy.

10.2 - wyznaczającą minimalną odległość między elementami o maksymalnej wartości(*uwaga: minimalna wartość w tablicy jest jedna, ale może wystąpić kilkakrotnie*).

Np. dla tablicy {9, 4, 9, 1, 7, 2, 9} maksymalna wartość wynosi 9 i występuje 3 razy. Minimalna odległość miedzy nimi wynosi 2 (różnica między indeksami 2 - trzeci element i 0 – pierwszy element). Jeśli wartość maksymalna występuje jednokrotnie to minimalna odległość wynosi 0.

11. Dana jest nieuporządkowana tablica jednowymiarowa (wektor) o długości *n*. Tablica ta zawiera tylko dwa rodzaje elementów. Posortuj tą tablicę w ten sposób, aby złożoność obliczeniowa algorytmu była jak najmniejsza. Sortowanie wykonaj za pomocą przestawiania odpowiednich elementów i bez używania pomocniczej tablicy. Np. jeśli tablica jest typu całkowitego i zawiera tylko elementy o wartości 0 i 1 , {0,1,1,0, ...0,1,0} to po uporządkowaniu jest postaci {0,0,0,........,1,1,1}.

12. Proszę napisać algorytm sprawdzający czy w tablicy jednowymiarowej połowy tablic są palindromami. To znaczy czy czytane od początku i od końca dają identyczny wynik. W przypadku tablicy o nieparzystym wymiarze pomijany jest element środkowy.
Np. A={$\boxed{1,2,2,1}$,$\boxed{3,4,1,3}$} nie spełnia warunków. Jej rozmiar wynosi 8. Podtablica o długości 4 od indeksu 0 do indeksu 3 jest palindromem, natomiast podtablica od ineksu 4 do indeksu 7 nie jest palindromem. Natomiast dla tablicy postaci A={$\boxed{1,2,3,2,1}$,0,$\boxed{5,5,9,5,5}$} spełnione są warunki. Rozmiar tablicy to 11. Dwie podtablice o wymiarach 5 są palindromami(jedna od indeksu 0 do 4, druga od indeksu 6 do 10). Element środkowy o indeksie 5 jest pomijany.

13.Proszę napisać algorytm, który sprawdza czy dwie tablice o wartościach całkowitych są anagramami, tzn. czy w każdej z nich można poprzestawiać tak elementy, aby była identyczna z drugą. Np. A={1,1,2,4,4}, B={1,2,2,4,5} nie są anagramami, natomiast A={1,2,2,6,1) i B={2,6,1,1,2} są.

14. Proszę napisać algorytm znajdujący w tablicy jednowymiarowej:

a) dominantę – wartość elementu w tablicy, występującego najczęściej (o największej krotności). Np. dla tablicy A={ 1,2,2,3,6,3,1,2,8,2,4,2,9,2} taką wartością jest 2.

b) Medianę – środkowy element w tablicy po jej uporządkowaniu. Jeśli tablica ma nieparzystą liczbę elementów, to ma jeden środkowy element. W przypadku parzystej liczby elementów, za medianę przyjmuje się jeden z dwóch środkowych elementów.
Np. dla A={1,4,8,6,3,1,9} medianą jest wartość 4. Dla B={2,3,8,4} za medianę można przyjąć albo 3 albo 4.

15. Proszę napisać funkcje wyznaczania wartości wielomianu jednorodnego stopnia *n* :

- w sposób iteracyjny minimalizując liczbę operacji,
- w sposób rekurencyjny minimalizując liczbe wywołan..

Wielomian stopnia *n* jest postaci:
$W_n(x)=a_0x^n+a_1x^{n-1}+a_2x^{n-2}+.......+a_{n-1}x^1+a_nx^0$.
Współczynniki wielomianu są dane w tablicy o wymiarze *n+1*.
Np.{ a_0, a_1, ,a_{n-1}, a_n}.

16. Proszę napisać funkcję wyznaczającą wartość $\mathbf{x}^n$ **(*n*>=0)** w sposob**:**
6.1 iteracyjny.
6.2 rekurencyjny na dwa sposoby: naiwny i przez podział.
Proszę porównać złożoności algorytmów.

17. Proszę napisać algorytm, który dla uporządkowanej niemalejąco tablicy jednowymiarowej, wyznacza krotność wystąpienia zadanej wartości. Uwaga: do tego celu należy między innymi użyć algorytmu wyszukiwania binarnego.

.

18. Proszę napisać funkcję, która dla tablicy jednowymiarowej o rozmiarze *n*, wyznacza punkt podziału tej tablice na dwie podtablice spełniające własność: minimalna wartość lewej podtablicy (od indeksu 0 do indeksu *k*) jest równa maksymalnej wartości prawej podtablicy(tej od indeksu *k+1* do *n-1*). Jeśli istnieje taki podział, funkcja zwraca punkt podziału (indeks *k*). W przeciwnym przypadku zwraca wartość –1.

0 1 2 3 | 4 5 6 7
Np. dla tablicy A={**5, 4, 6, 7, | 3,-4, 4,0** } takim punktem podziału jest indeks k=3. Wartość minimalna lewej podtablicy od indeksu 0 do 3 wynosi 4. Maksymalna wartość w prawej podtablicy od indeksu 4 do 7 wynosi również 4.

19 Proszę napisać algorytm, który dla tablicy jednowymiarowej o rozmiarze *n*, wyznacza punkt podziału tej tablice na dwie podtablice spełniające własność: suma wartości elementów w lewej podtablicy (od indeksu 0 do indeksu *k*) jest równa sumie wartości elementów prawej podtablicy(tej od indeksu *k+1* do *n-1*). Jeśli istnieje taki podział, funkcja zwraca punkt podziału (indeks *k*). W przeciwnym przypadku zwraca –1.

0 1 2 3 | 4 5 6 7
Np. dla tablicy A={7,-7, 5, 4| 3, 2, 4, 0} takim punktem podziału jest indeks k=3. Suma wartości elementów lewej podtablicy od indeksu 0 do 3 wynosi 9. Suma wartości w prawej podtablicy od indeksu 4 do 7 wynosi również 9.

20. Proszę napisać algorytm, który dla tablicy jednowymiarowej o rozmiarze **2*n*** **(**liczba parzysta), wyznacza pierwszy indeks podtablicy ciągłej o długości ***n*** spełniającej własność: suma elementów tej podtablicy wynosi połowę sumy elementów całej tablicy. Jeśli takiej podtablicy nie ma, wynikiem jest wartość –1.

0 1 2 3 4 5 6 7
Np. dla tablicy A={3, 2, 5, 4, 7, -7, 4, 0} suma wszystkich elementów wynosi 18. Rozmiar tablicy wynosi 8. Suma podtablicy o rozmiarze 4, od indeksu 2 do indeksu 5, wynosi 9.

21.Proszę napisać algorytm, który dla dwuwymiarowej tablicy wyznacza maksymalne pole kwadratu o zadanym rozmiarze. Szukany kwadrat leży wewnątrz tablicy. Jego pole to suma elementów tablicy zawartych w tym kwadracie.

22. Proszę napisać algorytm, który wypełnia „jodełką” tablicę kwadratowa o wymiarze n. Np. dla n równego 5, tablica ta jest postaci:

1 1 1 1 1
1 2 2 2 2
1 2 3 3 3
1 2 3 4 4
1 2 3 4 5

23. Proszę napisać algorytm, który dla dwuwymiarowej tablicy o wartościach całkowitych wyznacza numer wiersza o maksymalnej i nieparzystej wartości sumy elementów wiersza. Jeśli nie ma rozwiązania to wartością indeksu jest –1.
Np. dla tablicy o 4 wierszach i 3 kolumnach

1 5 1
2 3 3
2 2 5
4 3 3

będzie to indeks 2. W wierszach: o indeksie 0 suma wynosi 7, o indeksie 1 suma wynosi 8, o indeksie 2 suma wynosi 9, o indeksie 3 suma wynosi 10. Są dwa wiersze z sumą nieparzystą. Wartości tych sum to 7 i 9. Szukana wartość to 2. Jest to indeks wiersza z wartością sumy 9.

24. Proszę napisać algorytm, który zamienia w tablicy kwadratowej o wymiarze n, elementy nad przekątną z elementami pod przekątną. Ta zamiana to odbicie zwierciadlane względem przekątnej.
Np. dla n równego 5, tablica ta jest postaci:

1 2 3 4 *5*	*to po modyfikacji*	**1 6 6 6 6**
6 1 7 8 9		**2 1 2 2 2**
6 2 1 3 3		**3 7 1 4 5**
6 2 4 1 9		**4 8 3 1 0**
6 2 5 0 1		**5 9 3 9 1**

25. Proszę napisać algorytm, który dla dwuwymiarowej sprawdza czy minimalna wartość elementów tablicy nad główną przekątną jest równa maksymalnej wartości elementów pod główną przekątną. Jeśli warunek jest spełniony to wynikiem jest ta wartość. W przeciwnym wypadku wynikiem jest suma elementów z głównej przekątnej.
Np. dla tablicy o 4 wierszach i 4 kolumnach

1 5 1 1
2 3 3 2
2 2 2 2
4 3 5 2

wynikiem jest wartość 1+3+2+2 ponieważ minimum nad przekątną wynosi 1 a maksimum pod przekątną wynosi 5.

26. Proszę napisać algorytm wypełniający tablicę dwuwymiarową kwadratową w „paski" symetrycznie nad i pod przekątną prawa góra – lewy dół. Na przekątnej wartości 0. Na kolejnych „paskach" pod i nad przekątną wartości kolejno o jeden większe.
Np. dla tablicy dwuwymiarowej o rozmiarze 7 tablica ma postać:

6 5 4 3 2 1 0
5 4 3 2 1 0 1
4 3 2 1 0 1 2
3 2 1 0 1 2 3
2 1 0 1 2 3 4
1 0 1 2 3 4 5
0 1 2 3 4 5 6

27. Ciąg Fibonacciego zdefiniowany jest następująco:

$$Fib(n) = \begin{cases} 0, dla\, n = 0 \\ 1, dla\, n = 1 \\ Fib(n-2) + Fib(n-1), dla\, n > 1 \end{cases}$$

Dowolną liczbę naturalną możemy przedstawić w postaci ciągu zerojedynkowego zdefiniowanego w sposób następujący: jeżeli k-ta cyfra jest równa 1, wtedy odpowiada wartości Fib(k), jeżeli 0 to jest równa 0 (cyfry numerujemy od prawej do lewej zaczynając od jeden).
Przykład: $11011101_{\mathbf{Fib}}$ = 1(21) + 1(13) + 0(8) + 1(5) + 1(3) + 1(2) + 0(1) + 1(1) = 21 + 13 + 5 + 3 + 2 + 1 = $45_{\mathbf{10}}$
Napisz algorytm wyznaczający wartości dziesiętne takich ciągów zerojedynkowych.
Uwaga: elementy ciągu Fibonacciego nie mogą być stablicowane.

28. Algorytm liczący odwrotność modulo (rozszerzony algorytm Euklidesa).
Dla danych liczb naturalnych a i b znaleźć taką liczbę naturalną x, aby (a · x) mod b =1 lub stwierdzić, iż liczba x nie istnieje.

.

29. Algorytm wyszukujący najdłuższy palindrom w zerojedynkowym ciągu n - elementowym. W rozwiązaniu wystarczy podać długość palindromu i pozycję w ciągu rozpoczynającą wyszukaną sekwencję.
Przykład: dla 01101100110100110011101101110011001011011101000111100100011, najdłuższym palindromem jest 0110100110011101101110011001011O

30. Proszę napisać algorytm wyszukujący najkrótszą drogę w tablicy dwuwymiarowej od lewego, górnego rogu do prawego dolnego. Elementami tablicy są liczby naturalne, a dozwolone ruchy po tablicy to: → oraz ↓ .
Przykład: dla tablicy o 4 wierszach i 4 kolumnach zdefiniowanej poniżej, najkrótsza droga wynosi 13. Najkrótsza droga oznaczona jest na czerwono.

2	**0**	**3**	**2**
4	**4**	**1**	**2**
11	**12**	**15**	**4**
1	**5**	**3**	**1**