Imię i nazwisko:... **WdI, Egzamin, 2.02.2018**
Indeks:..

1. [20] *Rangą elementu* x[i] w ciągu liczb x[0], …, x[n – 1] nazywamy liczbę elementów tego ciągu mniejszych od x[i].

   Napisz funkcję (lub algorytm) realizującą następującą specyfikację:

   Wejście:
   n – liczba dodatnia parzysta,
   x – tablica liczb całkowitych taka, że x[0] ≤ x[1] ≤…≤ x[n/2 – 1] oraz
   x[n/2] ≤ x[n/2 + 1] ≤…≤ x[n – 1]

   Wyjście: ciąg liczb r[0], …, r[n – 1], gdzie r[i] jest rangą x[i] w ciągu x[0], …, x[n – 1].

   Opisz ideę swojego rozwiązania, uzasadnij poprawność i oszacuj złożoność czasową.

   Przykład
   Dla ciągu n = 4 i x[0], …, x[3] = [–1, 5, –4 , 7] Twoja funkcja-algorytm powinna zwrócić 1, 2, 0, 3.
   Dla ciągu n = 4 i x[0], …, x[3] = [–1, 5, –4 , 5] Twoja funkcja-algorytm powinna zwrócić 1, 2, 0, 2.
   Zauważ, że n = 4 i x[0], …, x[3] = [–1, 5, 7, –4] nie stanowią wejścia zgodnego ze specyfikacją.

   Uwaga:
   Maksymalna liczba punktów do uzyskania za to zadanie zależy od złożoności czasowej Twojego rozwiązania:
   - złożoność O(n^2): 8 punktów,
   - złożoność O(n log n): 15 punktów,
   - złożoność O(n): 20 punktów.

2. [20] Rozważmy następującą funkcję:

```
int z2(int n, int k){
    if (k>=n)
        return 1;
    if (n % k==0 && k>1)
        return 1+z2(n/k, k);
    return z2(n,k+1);
}
```

```
def z2(n,k):
    if k>=n:
        return 1
    if n % k==0 and k>1:
        return 1+z2(n/k, k)
    return z2(n,k+1)
```

Twoje zadanie:

a) [5] Prześledź działanie funkcji `z2` dla podanych poniżej wartości argumentów *n* i *k*. Uzupełnij brakujące wartości w ostatniej kolumnie tabeli.

n	*k*	*z2*(*n*, *k*)
10	1	2
10	2	
30	1	
72	1	
2048	1	
2048	4	

b) [5] Uzupełnij specyfikację poniższej funkcji `z2B`. Twoja specyfikacja powinna możliwie dokładnie opisywać działanie funkcji.

```
int z2B(int n){
  return z2(n, 1)
}
```

```
def z2B(n):
    return z2(n, 1)
```

Wejście: *n* – liczba naturalna większa od 1.
Wyjście:

c) [10] Napisz funkcję (lub algorytm) zgodną z poniższą specyfikacją.

Wejście: *n* – liczba naturalna większa od 1.
Wyjście: liczba różnych dzielników *n* będących liczbami pierwszymi.

Twoje rozwiązanie może (nie musi) działać podobnie do funkcji `z2`.

Przykład
Dla $n = 200 = 2^3 \cdot 5^2$ oraz dla $n = 10 = 2 \cdot 5$ należy zwrócić wartość 2, dla $n = 210 = 2 \cdot 3 \cdot 5 \cdot 7$ należy zwrócić wartość 4.

Imię i nazwisko:... **WdI, Egzamin, 2.02.2018**
Indeks:..

3. [20] Dane są następujące deklaracje:

```
typedef struct node *pnode;         class TreeItem:
typedef struct node{                def __init__(self,value):
        int val;                        self.val = value
        pnode left;                     self.left = None
        pnode right;} snode;            self.right = None
```

Drzewo o korzeniu *r* nazywamy *s-odseparowanym*, jeśli dla każdego węzła *v* tego drzewa zachodzi jeden z warunków:

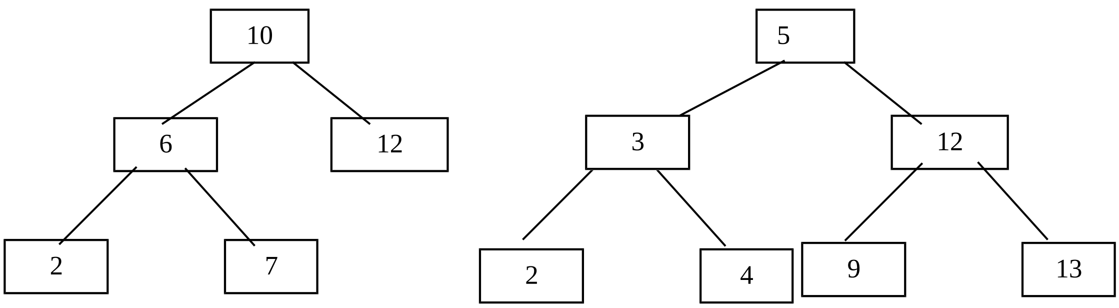
i. *v* ma co najwyżej jedno dziecko,
ii. $|S_{lewa} - S_{prawa}| \le s$, gdzie S_{lewa} to suma wartości elementów w lewym poddrzewie *v* a S_{prawa} to suma wartości elementów w prawym poddrzewie *v*.

Twoje zadanie:
(a) Napisz funkcję realizującą następującą specyfikację:

Wejście: r – korzeń drzewa (typu pnode lub TreeItem),
s – liczba całkowita.
Wyjście:
1 – jeśli drzewo o korzeniu r jest s-odseparowane,
0 – jeśli drzewo o korzeniu r **nie** jest s-odseparowane.

(b) Podaj asymptotyczny czas działania swojego rozwiązania wraz z uzasadnieniem.

Przykład
Dla drzewa o korzeniu 10 (po lewej): dla s=5 należy zwrócić wartość 1, natomiast dla s=4 należy zwrócić wartość 0. Dla drzewa o korzeniu 5 (po prawej): dla każdego s<25 należy zwrócić wartość 0, dla każdego s≥25 należy zwrócić wartość 1.



Uwaga:
Maksymalna liczba punktów do uzyskania za to zadanie zależy od złożoności czasowej Twojego rozwiązania:

– złożoność O(*n*): 20 punktów;
– rozwiązanie o złożoności większej niż O(*n*): 10 punktów.

4. **[10]** Rozważmy gramatykę bezkontekstową G(N, T, P, S), gdzie
N = {S},
T = { a, b, c, d } a zbiór P składa się z produkcji:

$S \rightarrow a\,S\,b$
$S \rightarrow c\,S\,d$
$S \rightarrow \varepsilon$
$S \rightarrow S\,S$

Uwaga: ε to słowo puste.

a) [5] Dla każdego z poniższych napisów podaj jego wyprowadzenie w gramatyce G lub uzasadnij, że nie należy do języka L(G):

a c b d

b a c d

a c d a b b

a c d a b b b

a c a b d b

b) [5] Wskaż, które z poniższych zdań są prawdziwe, a które fałszywe. Dla każdego z poniższych zdań zdecyduj, czy jest prawdziwe dla **<u>każdego</u>** napisu z języka L(G):

Zdanie	**Tak**	**Nie**
Liczba wystąpień znaku a jest większa niż liczba wystąpień znaku c		
W napisie nie występuje fragment abdc		
Napis nie zaczyna się od abcd		
Długość napisu jest parzysta		
Długość napisu jest podzielna przez 4		

Imię i nazwisko:... **WdI, Egzamin, 2.02.2018**
Indeks:..

5. **[30]** Rozważmy labirynt składający się z n pól ponumerowanych od 0 do n – 1, opisany przez tablice r1, r2. Z pola o numerze i można przejść do pól o numerach r1[i] i r2[i]. Wejście do labiryntu znajduje się w polu 0, a wyjście w polu n – 1. Labirynt nazywamy *przechodnim*, jeśli wychodząc z pola 0, można przejść do pola n – 1.

Zad. 5a. **[20]** Napisz funkcję (lub algorytm) realizującą następującą specyfikację:

Wejście: n – liczba naturalna,

r1, r2 – tablice liczb całkowitych, wypełnione liczbami z przedziału [0; n – 1]

Wyjście:
1 – jeśli labirynt opisany przez tablice r1, r2 jest przechodni,
0 – jeśli labirynt opisany przez tablice r1, r2 NIE jest przechodni.

Przykład
Dla $n = 5$, r1[*0..4*] = [*3, 0, 0, 0, 4*] i r2[*0..4*]=[*2, 3, 4, 1, 4*] funkcja powinna zwrócić wartość 1, gdyż zaczynając od pola 0, można przejść kolejno do 2 i 4.
Dla $n = 5$, r1[*0..4*] = [*3, 0, 0, 0, 4*] i r2[*0..4*]=[*2, 4, 3, 2, 4*] funkcja powinna zwrócić wartość 0, gdyż rozpoczynając od pola 0, nie da się przejść do pola 4.

Zad. 5b. [7] Labirynt jest *zachłannie przechodni*, gdy od 0 do n – 1 można dojść w ten sposób, że z pola i przechodzimy do pola max(r1[i], r2[i]). Napisz funkcję (lub algorytm) realizującą następującą specyfikację:

Wejście: n – liczba naturalna,

r1, r2 – tablice liczb całkowitych, wypełnione liczbami z przedziału [0; n – 1]

Wyjście:
1 – jeśli labirynt opisany przez tablice r1, r2 jest zachłannie przechodni,
0 – jeśli labirynt opisany przez tablice r1, r2 NIE jest zachłannie przechodni.

Zad. 5c. [3] Rozważmy labirynty, w których r1[i] $\geq i$ oraz r2[i] $\geq i$ dla każdego i.
Odpowiedz na pytanie: czy każdy labirynt przechodni spełniający powyższy warunek jest również labiryntem zachłannie przechodnim? Odpowiedź uzasadnij.