Zadanie 1. Test *(6 pkt)*

Zaznacz znakiem X w odpowiedniej kolumnie P lub F, która odpowiedź jest prawdziwa, a która fałszywa.

a) Przeanalizuj poniższy algorytm (:= oznacza instrukcję przypisania)

1. m:=0
2. n:=6
3. jeśli m>n to wykonaj krok 7.
4. m:=m+1
5. pisz m
6. przejdź do kroku 3.
7. stop

	P	F
Wykonywanie algorytmu zakończy się po wypisaniu liczb od 1 do 7.		
Po pierwszym sprawdzeniu warunku w kroku 3. nie zostaną wykonane kroki: 4., 5., 6. i wykonywanie algorytmu zakończy się.		
Wykonywanie algorytmu zakończy się po wypisaniu liczb od 0 do 6.		
Sprawdzenie warunku $m > n$ wykonane zostanie dokładnie 8 razy.		

Zadanie 2. Tablica zero-jedynkowa *(8 pkt)*

W tablicy a[1…1023] zapisano ciąg zer i jedynek w taki sposób, że wszystkie zera poprzedzają jedynki.

Uwaga: W tablicy mogą być same zera lub same jedynki.

Oto niepełny algorytm obliczania liczby zer w tablicy a:

$\leftarrow$ – oznacza instrukcję przypisania

div – oznacza dzielenie całkowite

```
liczba_zer ← 0
l ← 1, p ← 1023
dopóki l ≤ p wykonuj
        s ← (l + p) div 2
        jeśli a[s] = 1 to
              p ← s − 1
        w przeciwnym przypadku
              liczba_zer ← liczba_zer + ....................
              l ← ....................
```

a) Uzupełnij opis algorytmu, wstawiając w miejsce kropek stosowne wyrażenie, tak aby obliczał on zawsze poprawnie liczbę zer z tablicy a.

b) Ile instrukcji przypisania $s \leftarrow (l+p)\,div\,2$ jest wykonywanych w każdym przebiegu algorytmu? Odpowiedź uzasadnij.

Zadanie 1. Długość napisów binarnych *(7 pkt)*

Opisana poniżej funkcja rekurencyjna wyznacza, dla liczby naturalnej $n > 0$, długość napisu uzyskanego przez sklejenie binarnych reprezentacji liczb naturalnych od 1 do $n-1$.

Funkcja $sklej(n)$

krok 1. jeśli $n = 1$, to podaj 0 jako wynik i zakończ działanie

krok 2. jeśli n parzysta, to wynikiem jest $n-1+2 \cdot sklej(n/2)$

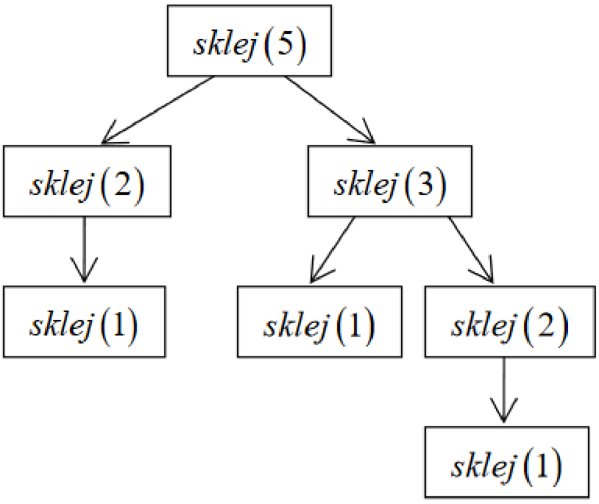
krok 3. jeśli n nieparzysta, to wynikiem jest $n-1+sklej((n-1)/2)+sklej((n+1)/2)$

Wykonaj polecenia a)–c):

a) Wykonanie funkcji *sklej* można przedstawić w postaci drzewa wywołań rekurencyjnych ilustrującego wszystkie wywołania funkcji po jej uruchomieniu dla zadanego argumentu. Poniższy rysunek przedstawia takie drzewo dla wywołania $sklej(5)$.

Wykonaj polecenia a)–c):

a) Wykonanie funkcji *sklej* można przedstawić w postaci drzewa wywołań rekurencyjnych ilustrującego wszystkie wywołania funkcji po jej uruchomieniu dla zadanego argumentu. Poniższy rysunek przedstawia takie drzewo dla wywołania $sklej(5)$.



Narysuj analogiczne drzewo dla wywołania $sklej(7)$.

b) Uzupełnij poniższą tabelę, podając wartości funkcji *sklej* dla wskazanych argumentów.

n	$sklej(n)$
1	0
2	1
3	
4	
5	
6	

Zadanie 1. Funkcja rekurencyjna *(8 pkt)*

Dana jest liczba naturalna $n > 0$ i tablica różnych liczb całkowitych $a[1..n]$. Rozważamy następującą **rekurencyjną** funkcję F z argumentem i będącym liczbą naturalną, $1 \le i \le n$.

Funkcja $F(i)$

jeżeli $i = n$ to
  wynikiem jest n
w przeciwnym razie
  $j := F(i+1)$
  jeżeli $a[i] < a[j]$ wtedy
    wynikiem jest i
  w przeciwnym razie
    wynikiem jest j

a) Dla danej 10-elementowej tablicy $a = [5, 1, 8, 9, 7, 2, 3, 11, 20, 15]$ podaj w poniższej tabeli wynik wywołania funkcji F dla danego argumentu i.

i	$F(i)$
9	
7	
5	

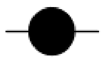
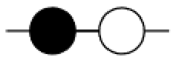
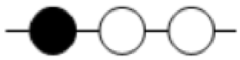
Zadanie 1. Korale *(8 pkt)*

Rozważamy następującą **rekurencyjną** procedurę ***Korale***, której parametrem jest dodatnia liczba całkowita *n*.

Korale(n)

1. Jeżeli *n* = 1, to
   1.1. nawlecz czarny koralik na prawy koniec sznurka,
   1.2. zakończ działanie procedury.
2. Jeżeli *n* jest parzyste, to
   2.1. wykonaj ***Korale(n/2)***,
   2.2. nawlecz biały koralik na prawy koniec sznurka,
   2.3. zakończ działanie procedury.
3. Jeżeli *n* jest nieparzyste, to
   3.1. wykonaj ***Korale((n-1)/2)***,
   3.2. nawlecz czarny koralik na prawy koniec sznurka,
   3.3. zakończ działanie procedury.

a) Uzupełnij tabelę i w ten sposób przedstaw wynik działania powyższego algorytmu dla podanych argumentów *n*:

n	wynik działania **Korale(*n*)**
1	
2	
3	
4	
7	
8	
15	
16	

Zadanie 1. Doskonała inaczej *(6 pkt)*

Poniższy algorytm wyznacza wszystkie dzielniki liczby naturalnej $n \geq 1$, mniejsze od *n*.

Specyfikacja algorytmu:

Dane: liczba naturalna $n \geq 1$,

Wynik: ciąg liczb, które są dzielnikami liczby *n*, mniejszymi od *n*.

Algorytm:

1. $d \leftarrow 1$
2. dopóki $d < n$ wykonuj
   - 2.1. jeżeli *n* mod $d = 0$, to wypisz *d*
   - 2.2. $d \leftarrow d+1$

Uwaga: „*n* mod *d*” oznacza resztę z dzielenia liczby *n* przez *d*, np. 5 mod 2 = 1, 6 mod 2 = 0.

a) Uzupełnij poniższą tabelę – podaj wyniki działania algorytmu dla wskazanych argumentów:

n	Wynik algorytmu
6	1 2 3
35	
56	
81	