

Pseudokody z ćwiczeń 1 i 2

Ćwiczenie 1. Dane są nieujemna liczba całkowita a oraz dodatnia liczba całkowita b . Nie stosując operacji *div* i *mod* podać algorytm wyznaczający iloraz i resztę z dzielenia a przez b .

ILORAZ(a, b)

$r = a$

$q = 0$

while $r \geq b$

do $r = r - b$

$q = q + 1$

return q, r

Ćwiczenie 2. Dane są liczby całkowite a oraz b . Wyznaczyć wartość wyrażenia $a * b$ nie stosując operacji mnożenia.

ILOCZYN(a, b)

if $b < 0$

then $b = -b$

$a = -a$

$c = 0$

while $b > 0$

do $c = c + a$

$b = b - 1$

return c

Ćwiczenie 3. Dla danej nieujemnej liczby całkowitej n oraz liczby naturalnej $0 < k \leq n$ zapisz w pseudokodzie ciąg instrukcji obliczający wartość wyrażenia

$$n(n-1) \dots (n-k+1)$$

WYRAZENIE(k, n)

$m = n - k + 1$

$w = m$

for $i = 1$ **to** $k - 1$

do $w = w * (m + i)$

return w

Ćwiczenie 4. Dana jest liczba naturalna n . Zapisz pseudokod algorytmu wypisującego wartości kolejnych kwadratów liczb naturalnych aż do n^2 i obliczający ich sumę, w którym jedyną dopuszczalną operacją arytmetyczną jest dodawanie.

SUMA(n)

$s = 0$

$k = 0$

$kw = 0$

while $k \neq n$

do $kw = kw + k$

$k = k + 1$

$kw = kw + k$

kw)

$s = s + kw$

return s

Ćwiczenie 5. Algorytm, który dla danej nieujemnej liczby całkowitej n oblicza sumę kolejnych odwrotności silni od 1 do n .

SILNIE(n)

$silnia = 1$

$s = 1$

for $k = 2$ **to** n

do $silnia = silnia * k$

$s = s + 1/silnia$

return s

Ćwiczenie 6. Algorytm, który wczytuje kolejno liczby dopóki nie pojawi się liczba n i jako wynik podaje liczbę elementów mniejszych bądź równych k .

WCZYTUJ(k, n)

$l = 0$

czytaj (a)

while $a \neq n$

do if $a \leq k$

then $l = l + 1$

 czytaj (a)

return l

Ćwiczenie 7. Algorytm, który wypisuje sumę iloczynów zadanej liczby rzeczywistej a przez wszystkie liczby parzyste od 2 do n .

SUMA(a, n)

$s = 0$

$k = 2$

while $k \leq n$

do $s = s + k$

$k = k + 2$

$s = a * s$

return s

Ćwiczenie 8. Dana jest tablica $A[1..n]$ zawierająca liczby całkowite, przy czym $A[1] \leq A[2] \leq \dots \leq A[n]$. Algorytm wyznaczający liczbę różnych elementów tej tablicy.

ROZNE(A, n)

```

 $k = 1$ 
for  $i = 1$  to  $n - 1$ 
    do if  $A[i] \neq A[i + 1]$ 
        then  $k = k + 1$ 
return  $k$ 

```

Ćwiczenie 9. Algorytm „odwracający” tablicę $A[1..n]$, tzn. pierwszy element staje się ostatnim a ostatni pierwszym, drugi element staje się przedostatnim itd. Wykorzystać instrukcję zamiany zmiennych $a \leftrightarrow b$.

```

ODW( $n, A$ )
 $m = n \text{ div } 2$ 
for  $i = 1$  to  $m$ 
    do  $A[i] \leftrightarrow A[n - i + 1]$ 
return  $A$ 

```