

Oświadczam, że podczas pisania tego kodu nie używałem pomocy zewnętrznych ani osób trzecich.

1.

a) Wyszukajcie najmniejszej ( $l$ ) i największej ( $h$ ) kolumny w tabeli oraz miejsca ( $j$ ) między najmniejszej i największej.

$$a=3$$

$$b=3$$

$$c=2$$

b)

$$i=5, a=7, l=2, j=2$$

2.

$$A[1..m, 1..n]$$

function <sup>zad2</sup> ( $A[1..m, 1..n], m, n$ )

wynik = 0

// zmienna do przechowywania wyniku

for  ~~$i=1$  to  $m$  do~~  $i=1$  to  $m$  do

for  ~~$j=1$  to  $n$  do~~  $j=1$  to  $n$  do // przejście przez całą tabelę

~~if  $(i=1 \vee i=m \vee j=1 \vee j=n)$  then~~

if  $(i=1 \vee i=m \vee j=1 \vee j=n)$  then // warunek na wypisanie

write  $A[i, j]$

wynik = wynik +  $(A[i, j] * A[i, j])$  // pierwszy i ostatni

elseif  $(A[i, j] \bmod 2 == 0)$  then // warunek na parzyste

write  $A[i, j]$

// elementy tabeli

wynik = wynik +  $(A[i, j] * A[i, j])$  // sumowanie ich

// kwadratów

return wynik

~~or - podwójna kreska oznacza zapis or~~

30

function podzielna9(n)

a = 0 // suma cyfr

while n  $\neq$  0 do // dopóki n różno od zera, pętla

a = a + n mod 10 // dodanie do a ostatniej cyfry

n = n div 10 // usunięcie ostatniej cyfry

if (a mod 9 == 0) then

~~the~~ return true

else ~~for~~ then

return false

return podzielna9(ABS(n))

// sprawdzenie czy suma jest

// podzielna przez 9

~~for~~ Procedure Powtarzajac(T[1..n], n

~~for~~ i := 1; i  $\leq$  n; i++ i := 1 to n do

~~for~~ B[i] := T[i]

// wypisanie do powtarzanej

~~for~~



Raquel Kalinowski 14.12.2020

1.1.

Procedure Powtorzenie( $T[1..n]$ ,  $n$ )

// zmieni powtórzenie

$a = 1, b = 0$

for  $i = 1$  to  $n$  do

~~for  $j = 1$  to  $n$  do  
 $T[i, j] = T[i, j] + a$~~

for  $j = i+1$  to  $n$  do

if  $T[i, j] = T[i, j]$

$B[a] = T[i, j]$

$b = b + 1$

$a = a + 1$

j := n

// wypisanie powtórzenia się znowu

// zakończenie pętli

for  $i = 1$  to  $a$  do

print ~~powtórzenie~~  $B[i]$

print

~~powtórzenie~~

// wypisanie powtórzenia się znowu

// wypisanie ilości powtórzeń

$$a) \quad T(n) = 5T\left(\frac{n}{3}\right) + n^2$$

$$a=5, \quad b=3, \quad f(n)=n^2$$

$$\log_3 5 < 2 \quad g(n) = n^{\log_3 5} < n^2$$

$$T(n) = \Theta(n^2)$$

$$b) \quad T(n) = 5T\left(\frac{n}{2}\right) + n^2$$

$$a=5, \quad b=2, \quad f(n)=n^2$$

$$\log_2 5 > 2 \quad \Rightarrow \quad g(n) = n^{\log_2 5} > f(n)$$

$$T(n) = \Theta(n^{\log_2 5})$$

$$c) \quad T(n) = 64T\left(\frac{n}{8}\right) + n^2$$

$$a=64, \quad b=8, \quad f(n)=n^2$$

$$\log_8 64 = 2 \quad \Rightarrow \quad g(n) = n^2 = f(n)$$

$$T(n) = \Theta(n^2 \cdot \log n)$$



6

function sortowanie (  $B[1..k, 1..k]$ ,  $k$  )

 $a = 0$ 

// zmiana do poszukiwania minimum

 $j = 1$ 

~~for  $i = 1$  to  $k$  do~~  
 ~~$a = B[i, j]$~~  ~~for  $j = 1$  to  $k$  do~~

for  $i = 1$  to  $k$  do

 $a = a \leftarrow B[i, 1]$ 

for  $j = 1$  to  $k$  do

~~$a = B[i, j]$~~

if  $a > B[i, j]$

 $A[i] = a$ 

else if

 $a = B[i, j]$  $A[i] = a$ 

// wywołanie funkcji A  
// najniższy w kolumnie znowu B

for  $i = 2$  to  $k$  do

 $j = i$ 

while  $A[j] > A[j-1]$  and  $j > 1$  do

 $A[j] \leftrightarrow A[j-1]$  // operacja swap $j = j - 1$ 

return  $A[1..k]$