

INSTRUKCJE ITERACYJNE

Zadanie nr 1

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który wyświetla na ekranie monitora 10 kolejnych liczb całkowitych począwszy od 1. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START i := 1 DOPÓKI i <= 10 WYKONUJ POCZĄTEK PISZ(i) i := i + 1 KONIEC STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1] Init --> Cond{i <= 10} Cond -- TAK --> Print[/PISZ(i)/] Print --> Inc[i := i + 1] Inc --> Cond Cond -- NIE --> Stop([STOP]) </pre> |
| <pre> START i := 1 POWTARZAJ PISZ(i) i := i + 1 AŻ i > 10 STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1] Init --> Print[/PISZ(i)/] Print --> Inc[i := i + 1] Inc --> Cond{i > 10} Cond -- TAK --> Stop([STOP]) Cond -- NIE --> Print </pre> |

Zadanie nr 2

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który oblicza sumę 10 kolejnych liczb całkowitych począwszy od 1 i wyświetla ją na ekranie monitora. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START i := 1 s := 0 DOPÓKI i <= 10 WYKONUJ POCZĄTEK s := s + i i := i + 1 KONIEC PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 s := 0] Init --> Cond{i <= 10} Cond -- TAK --> Body[s := s + i i := i + 1] Body --> Cond Cond -- NIE --> Print[/PISZ(s)/] Print --> Stop([STOP]) </pre> |
| <pre> START i := 1 s := 0 POWTARZAJ s := s + i i := i + 1 AŻ i > 10 PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 s := 0] Init --> Body[s := s + i i := i + 1] Body --> Cond{i > 10} Cond -- TAK --> Print[/PISZ(s)/] Print --> Stop([STOP]) Cond -- NIE --> Body </pre> |

Zadanie nr 3

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który oblicza sumę 10 kolejnych parzystych liczb całkowitych począwszy od 2 i wyświetla ją na ekranie monitora. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START i := 1 j := 2 s := 0 DOPÓKI i <= 10 WYKONUJ POCZĄTEK s := s + j j := j + 2 i := i + 1 KONIEC PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 j := 2 s := 0] Init --> Dec{i <= 10} Dec -- TAK --> Body[s := s + j j := j + 2 i := i + 1] Body --> Dec Dec -- NIE --> Print[/PISZ(s)/] Print --> Stop([STOP]) </pre> |
| <pre> START i := 1 j := 2 s := 0 POWTARZAJ s := s + j j := j + 2 i := i + 1 AŻ i > 10 PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 j := 2 s := 0] Init --> Body[s := s + j j := j + 2 i := i + 1] Body --> Dec{i > 10} Dec -- TAK --> Print[/PISZ(s)/] Print --> Stop([STOP]) Dec -- NIE --> Body </pre> |

Zadanie nr 4

Ojciec ma syna, któremu daje przez 30 dni pieniądze w następujący sposób. Pierwszego dnia syn otrzymuje 2 złote, każdego następnego dnia otrzymuje o dwa złote więcej niż w dniu poprzednim. Ile pieniędzy zaoszczędzi syn.

Odpowiedź

| pseudocode | block diagram |
|---|--|
| <pre> START i := 1 j := 2 s := 0 DOPÓKI i <= 30 WYKONUJ POCZĄTEK s := s + j j := j + 2 i := i + 1 KONIEC PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 j := 2 s := 0] Init --> Dec{i <= 30} Dec -- TAK --> Loop[s := s + j j := j + 2 i := i + 1] Loop --> Dec Dec -- NIE --> Print[/PISZ(s)/] Print --> Stop([STOP]) </pre> |
| <pre> START i := 1 j := 2 s := 0 POWTARZAJ s := s + j j := j + 2 i := i + 1 AŻ i > 30 PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 j := 2 s := 0] Init --> Loop[s := s + j j := j + 2 i := i + 1] Loop --> Dec{i > 30} Dec -- TAK --> Print[/PISZ(s)/] Print --> Stop([STOP]) Dec -- NIE --> Loop </pre> |

Zadanie nr 5

Ojciec ma syna, któremu daje przez 10 dni pieniądze w następujący sposób. Pierwszego dnia syn otrzymuje 2 złote, każdego następnego dnia otrzymuje dwa razy więcej niż w dniu poprzednim. Ile pieniędzy zaoszczędzi syn.

Odpowiedź

| pseudocode | block diagram |
|---|--|
| <pre> START i := 1 j := 2 s := 0 DOPÓKI i <= 10 WYKONUJ POCZĄTEK s := s + j j := j * 2 i := i + 1 KONIEC PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 j := 2 s := 0] Init --> Dec{i <= 10} Dec -- TAK --> Loop[s := s + j j := j * 2 i := i + 1] Loop --> Dec Dec -- NIE --> Print[/PISZ(s)/] Print --> Stop([STOP]) </pre> |
| <pre> START i := 1 j := 2 s := 0 POWTARZAJ s := s + j j := j * 2 i := i + 1 AŻ i > 10 PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 j := 2 s := 0] Init --> Loop[s := s + j j := j * 2 i := i + 1] Loop --> Dec{i > 10} Dec -- TAK --> Print[/PISZ(s)/] Print --> Stop([STOP]) Dec -- NIE --> Loop </pre> |

Zadanie nr 6

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który wyświetla na ekranie 20 elementów ciągu arytmetycznego: pierwszy element ciągu ma wartość 150, każdy element następny jest o 5 mniejszy. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START i := 1 j := 150 DOPÓKI i <= 20 WYKONUJ POCZĄTEK PISZ(j) j := j - 5 i := i + 1 KONIEC STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 j := 150] Init --> Cond{i <= 20} Cond -- TAK --> Print[/PISZ(j)/] Print --> Update[j := j - 5 i := i + 1] Update --> Cond Cond -- NIE --> Stop([STOP]) </pre> |
| <pre> START i := 1 j := 150 POWTARZAJ PISZ(j) j := j - 5 i := i + 1 AŻ i > 20 STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 j := 150] Init --> Print[/PISZ(j)/] Print --> Update[j := j - 5 i := i + 1] Update --> Cond{i > 20} Cond -- TAK --> Stop([STOP]) Cond -- NIE --> Print </pre> |

Zadanie nr 7

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który oblicza ilość cegieł w piramidzie o podstawie 10 na 10 cegieł. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START i := 1 s := 0 DOPÓKI i <= 10 WYKONUJ POCZĄTEK s := s + i*i i := i + 1 KONIEC PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 s := 0] Init --> Cond{i <= 10} Cond -- TAK --> Body[s := s + i*i i := i + 1] Body --> Cond Cond -- NIE --> Print[/PISZ(s)/] Print --> Stop([STOP]) </pre> |
| <pre> START i := 1 s := 0 POWTARZAJ s := s + i*i i := i + 1 AŻ i > 10 PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 s := 0] Init --> Body[s := s + i*i i := i + 1] Body --> Body Body --> Cond{i > 10} Cond -- TAK --> Print[/PISZ(s)/] Print --> Stop([STOP]) Cond -- NIE --> Body </pre> |

Zadanie nr 8

Pan Kowalski zbudował ścianę z cegieł. Każda cegła ważyła 2 kilogramy. U podstawy ściany znajdowało się 15 cegieł. Każda warstwa wyższa posiadała o jedną cegłę mniej. Napisz algorytm za pomocą pseudojęzyka i schematów blokowych, który oblicza masę ściany. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|--|--|
| <pre> START i := 15 s := 0 DOPÓKI i >= 1 WYKONUJ POCZĄTEK s := s + i*i i := i - 1 KONIEC s := s*2 PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 15 s := 0] Init --> Dec{i >= 1} Dec -- TAK --> Loop[s := s + i*i i := i - 1] Loop --> Dec Dec -- NIE --> Calc[s := s*2] Calc --> Print[/PISZ(s)/] Print --> Stop([STOP]) </pre> |
| <pre> START i := 15 s := 0 POWTARZAJ s := s + i*i i := i - 1 AŻ i < 1 s := s*2 PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 15 s := 0] Init --> Loop[s := s + i*i i := i - 1] Loop --> Loop Loop --> Dec{i < 1} Dec -- TAK --> Calc[s := s*2] Calc --> Print[/PISZ(s)/] Print --> Stop([STOP]) Dec -- NIE --> Loop </pre> |

Zadanie nr 9

Napisz algorytm za pomocą pseudojęzyka i schematów blokowych, który wyświetla na ekranie monitora tabliczkę mnożenia w następujący sposób:

- Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.
- Wartość pierwszego czynnika w każdym działaniu wynosi 5
- Wartość drugiego czynnika w działaniu pierwszym ma wartość 3 a w każdym następnym jest o jeden większa.
- Program ma wyświetlać 20 działań

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START i := 3 DOPÓKI i <= 22 WYKONUJ POCZĄTEK PISZ(5;"*" ;i;"=" ;5*i) i := i + 1 KONIEC STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 3] Init --> Cond{i <= 22} Cond -- TAK --> Print[/PISZ(5;"*" ;i;"=" ;5*i)/] Print --> Inc[i := i + 1] Inc --> Cond Cond -- NIE --> Stop([STOP]) </pre> |
| <pre> START i := 3 POWTARZAJ PISZ(5;"*" ;i;"=" ;5*i) i := i + 1 AŻ i > 22 STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 3] Init --> Print[/PISZ(5;"*" ;i;"=" ;5*i)/] Print --> Inc[i := i + 1] Inc --> Cond{i > 22} Cond -- TAK --> Stop([STOP]) Cond -- NIE --> Print </pre> |

Zadanie nr 10

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który wyświetla na ekranie monitora ciąg n liczb całkowitych w którym każda liczba jest większa od poprzedniej o określoną wartość. Wartość pierwszego elementu ciągu m , ilość elementów ciągu n i różnica między elementami ciągu r jest podawana przez użytkownika podczas działania algorytmu. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|--|---|
| <pre> START PODAJ(m;n;r) i:=1 DOPÓKI i <= n WYKONUJ POCZĄTEK PISZ(m) m := m + r i := i + 1 KONIEC STOP </pre> | <pre> graph TD Start([START]) --> Input[/PODAJ(m;n;r)/] Input --> Init[i := 1] Init --> Cond{i <= n} Cond -- TAK --> Print[/PISZ(m)/] Print --> Calc[m := m + r i := i + 1] Calc --> Cond Cond -- NIE --> Stop([STOP]) </pre> |
| <pre> START PODAJ(m;n;r) i := 1 POWTARZAJ PISZ(m) m := m + r i := i + 1 AŻ i > n STOP </pre> | <pre> graph TD Start([START]) --> Input[/PODAJ(m;n;r)/] Input --> Init[i := 1] Init --> Print[/PISZ(m)/] Print --> Calc[m := m + r i := i + 1] Calc --> Cond{i > n} Cond -- TAK --> Stop([STOP]) Cond -- NIE --> Print </pre> |

Zadanie nr 11

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który ma obliczać sumę dwóch liczb a oraz b których wartości są podawane przez użytkownika oraz wyświetlać jej wartość na ekranie monitora. Wartość liczb są podawane dopóki ich suma jest różna od zera. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START a:=1 b:=1 DOPÓKI a+b <> 0 WYKONUJ POCZĄTEK PODAJ(a;b) c := a + b PISZ(c) KONIEC STOP </pre> | <pre> graph TD Start([START]) --> Init[a := 1 b := 1] Init --> Dec{a+b <> 0} Dec -- TAK --> Podaj[/PODAJ(a;b)/] Podaj --> Calc[c := a+b] Calc --> Pisz[/PISZ(c)/] Pisz --> Dec Dec -- NIE --> Stop([STOP]) </pre> |
| <pre> START POWTARZAJ PODAJ(a;b) c := a + b PISZ(c) AŻ a+b = 0 STOP </pre> | <pre> graph TD Start([START]) --> Podaj[/PODAJ(a;b)/] Podaj --> Calc[c := a+b] Calc --> Pisz[/PISZ(c)/] Pisz --> Dec{a+b = 0} Dec -- TAK --> Stop([STOP]) Dec -- NIE --> Podaj </pre> |

Zadanie nr 12

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który ma realizować następujące założenia:

- program wczytuje z klawiatury wartości trzech zmiennych **a** (pierwszy element ciągu), **n** (ilość elementów ciągu), **r** (różnica między elementami ciągu),
- algorytm oblicza sumę oraz iloczyn tego ciągu.
- Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|-----------------|
| <p>START PODAJ(a;n;r) i:=1 s:=0 p :=1 DOPÓKI i <= n WYKONUJ POCZĄTEK s := s + a p := p * a a := a + r i := i + 1 KONIEC PISZ(s;p) STOP</p> <p>Wyjaśnienie pojęć: s – suma p – iloczyn</p> | |
| <p>START PODAJ(a;n;r) i:=1 s:=0 p :=1 POWTARZAJ s := s + a p := p * a a := a + r i := i + 1 AŻ i > n PISZ(s;p) STOP</p> <p>Wyjaśnienie pojęć: s – suma p – iloczyn</p> | |

Zadanie nr 13

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który oblicza silnię podanej liczby i wyświetla jej wartość na ekranie monitora. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START PODAJ(n) i := 1 p := 1 DOPÓKI i <= n WYKONUJ POCZĄTEK p := p * i i := i + 1 KONIEC PISZ(p) STOP </pre> | <pre> graph TD Start([START]) --> Input[/PODAJ(n)/] Input --> Init[i := 1 p := 1] Init --> Cond{i <= n} Cond -- TAK --> Proc[p := p * i i := i + 1] Proc --> Cond Cond -- NIE --> Output[/PISZ(p)/] Output --> Stop([STOP]) </pre> |
| <pre> START PODAJ(n) i := 1 p := 1 POWTARZAJ p := p * i i := i + 1 AŻ i > n PISZ(p) STOP </pre> | <pre> graph TD Start([START]) --> Input[/PODAJ(n)/] Input --> Init[i := 1 p := 1] Init --> Proc[p := p * i i := i + 1] Proc --> Proc Proc --> Cond{i > n} Cond -- TAK --> Output[/PISZ(p)/] Output --> Stop([STOP]) Cond -- NIE --> Proc </pre> |

Zadanie nr 14

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który oblicza silnię podanej liczby oraz sumę silni poszczególnych liczb od 1 do tej liczby i wyświetla ich wartości na ekranie monitora. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START PODAJ(n) i := 1 p := 1 s := 0 DOPÓKI i <= n WYKONUJ POCZĄTEK p := p * i s := s + p i := i + 1 KONIEC PISZ(s;p) STOP </pre> | <pre> graph TD Start([START]) --> Input[/PODAJ(n)/] Input --> Init[i := 1 p := 1 s := 0] Init --> Cond{i <= n} Cond -- TAK --> Loop[p := p * i s := s + p i := i + 1] Loop --> Cond Cond -- NIE --> Output[/PISZ(s;p)/] Output --> Stop([STOP]) </pre> |
| <pre> START PODAJ(n) i := 1 p := 1 s := 0 POWTARZAJ p := p * i s := s + p i := i + 1 AŻ i > n PISZ(s;p) STOP </pre> | <pre> graph TD Start([START]) --> Input[/PODAJ(n)/] Input --> Init[i := 1 p := 1 s := 0] Init --> Loop[p := p * i s := s + p i := i + 1] Loop --> Cond{i > n} Cond -- TAK --> Output[/PISZ(s;p)/] Output --> Stop([STOP]) Cond -- NIE --> Loop </pre> |

Zadanie nr 15

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który wypisuje ciąg n liczb w którym każda liczba jest równa liczbie poprzedniej podniesionej do kwadratu. Pierwsza liczba jest podawana przez użytkownika. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|--|--|
| <pre> START PODAJ(a;n) i := 1 DOPÓKI i <= n WYKONUJ POCZĄTEK PISZ(a) a := a * a i := i + 1 KONIEC STOP </pre> | <pre> graph TD Start([START]) --> Input[/PODAJ(a;n)/] Input --> Init[i := 1] Init --> Dec{i <= n} Dec -- TAK --> Print[/PISZ(a)/] Print --> Calc[a := a * a i := i + 1] Calc --> Dec Dec -- NIE --> Stop([STOP]) </pre> |
| <pre> START PODAJ(a;n) i := 1 POWTARZAJ PISZ(a) a := a * a i := i + 1 AŻ i > n STOP </pre> | <pre> graph TD Start([START]) --> Input[/PODAJ(a;n)/] Input --> Init[i := 1] Init --> Print[/PISZ(a)/] Print --> Calc[a := a * a i := i + 1] Calc --> Dec{i > n} Dec -- TAK --> Stop([STOP]) Dec -- NIE --> Print </pre> |

Zadanie nr 16

Przedstaw algorytm za pomocą pseudojęzyka i schematów blokowych, który ma obliczać sumę 20 losowych liczb całkowitych z przedziału $\langle 30;60 \rangle$. Zrealizuj problem za pomocą instrukcji DOPÓKI i POWTARZAJ.

Odpowiedź

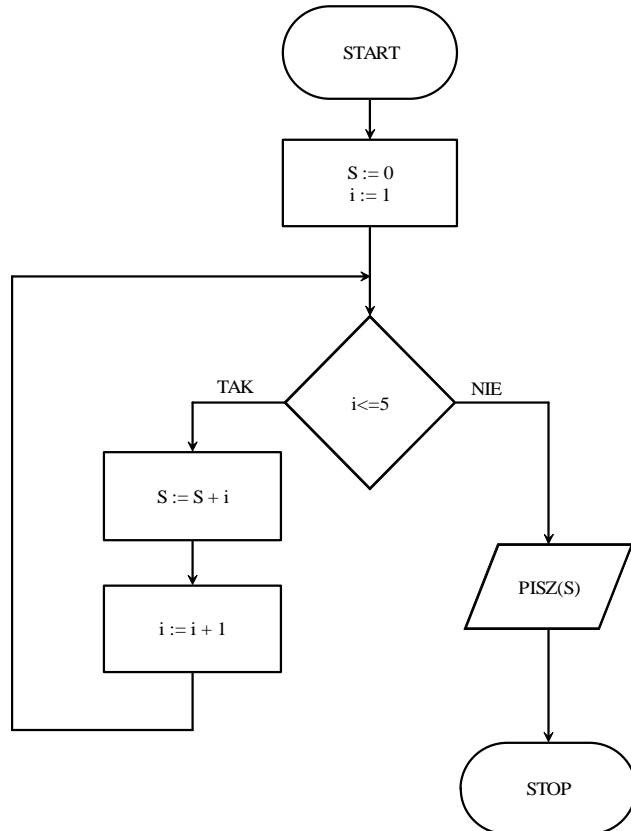
| pseudojęzyk | schemat blokowy |
|---|--|
| <pre> START i := 1 s := 0 DOPÓKI i <= n WYKONUJ POCZĄTEK x := LOSOWA(30;60) s := s + x i := i + 1 KONIEC PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 s := 0] Init --> Cond{i <= n} Cond -- TAK --> Body[x := LOSOWA(30;60) s := s + x i := i + 1] Body --> Cond Cond -- NIE --> Print[/PISZ(s)/] Print --> Stop([STOP]) </pre> |
| <pre> START i := 1 s := 0 POWTARZAJ x := LOSOWA(30;60) s := s + x i := i + 1 AŻ i > n PISZ(s) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 s := 0] Init --> Body[x := LOSOWA(30;60) s := s + x i := i + 1] Body --> Cond{i > n} Cond -- TAK --> Print[/PISZ(s)/] Print --> Stop([STOP]) Cond -- NIE --> Body </pre> |

Zadanie nr 17

Obok przedstawiono schemat blokowy. Ułóż treść zadania do tego schematu.

Odpowiedź

Przedstaw algorytm za pomocą schematów blokowych, który oblicza sumę 5 kolejnych liczb całkowitych począwszy od 1 i wyświetla ją na ekranie monitora. Zrealizuj problem za pomocą instrukcji DOPÓKI.



Zadanie nr 18

Obok przedstawiono schemat blokowy. Wskaż błąd w tym schemacie oraz ułóż do niego treść zadania.

Odpowiedź

| Tak jest | Tak powinno być |
|---|---|
| <pre> graph TD Start([START]) --> Init[S := 0 i := 1 j := 12] Init --> Dec{j <= 10} Dec -- TAK --> Sum[S := S + j] Sum --> Inc[i := i + 1 j := j + 2] Inc --> Dec Dec -- NIE --> Print[/PISZ(S)/] Print --> Stop([STOP]) </pre> | <pre> graph TD Start([START]) --> Init[S := 0 i := 1 j := 12] Init --> Dec{i <= 10} Dec -- TAK --> Sum[S := S + j] Sum --> Inc[i := i + 1 j := j + 2] Inc --> Dec Dec -- NIE --> Print[/PISZ(S)/] Print --> Stop([STOP]) </pre> |
| <p>Wyjaśnienie: Warunek ma postać $j \leq 10$ a powinien mieć postać $i \leq 10$</p> | |

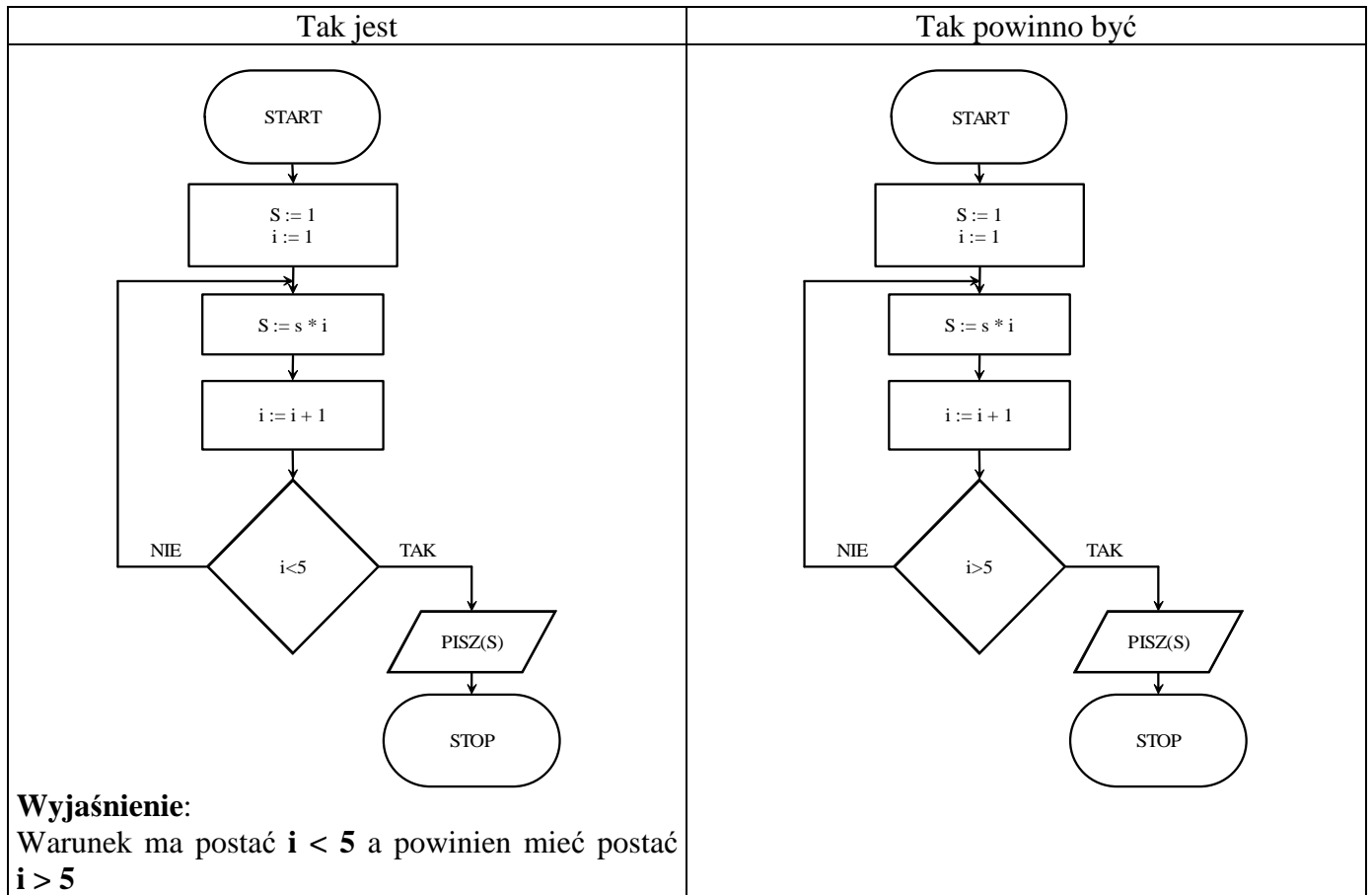
Przykładowa treść zadania:

Napisz algorytm za pomocą schematów blokowych, który oblicza sumę 10 elementów ciągu arytmetycznego: 12, 14 ..., wyświetl wynik na ekranie monitora. Zrealizuj zadanie za pomocą instrukcji DOPÓKI.

Zadanie nr 19

Obok przedstawiono schemat blokowy. Wskaż błąd w tym schemacie oraz ułóż do niego treść zadania.

Odpowiedź



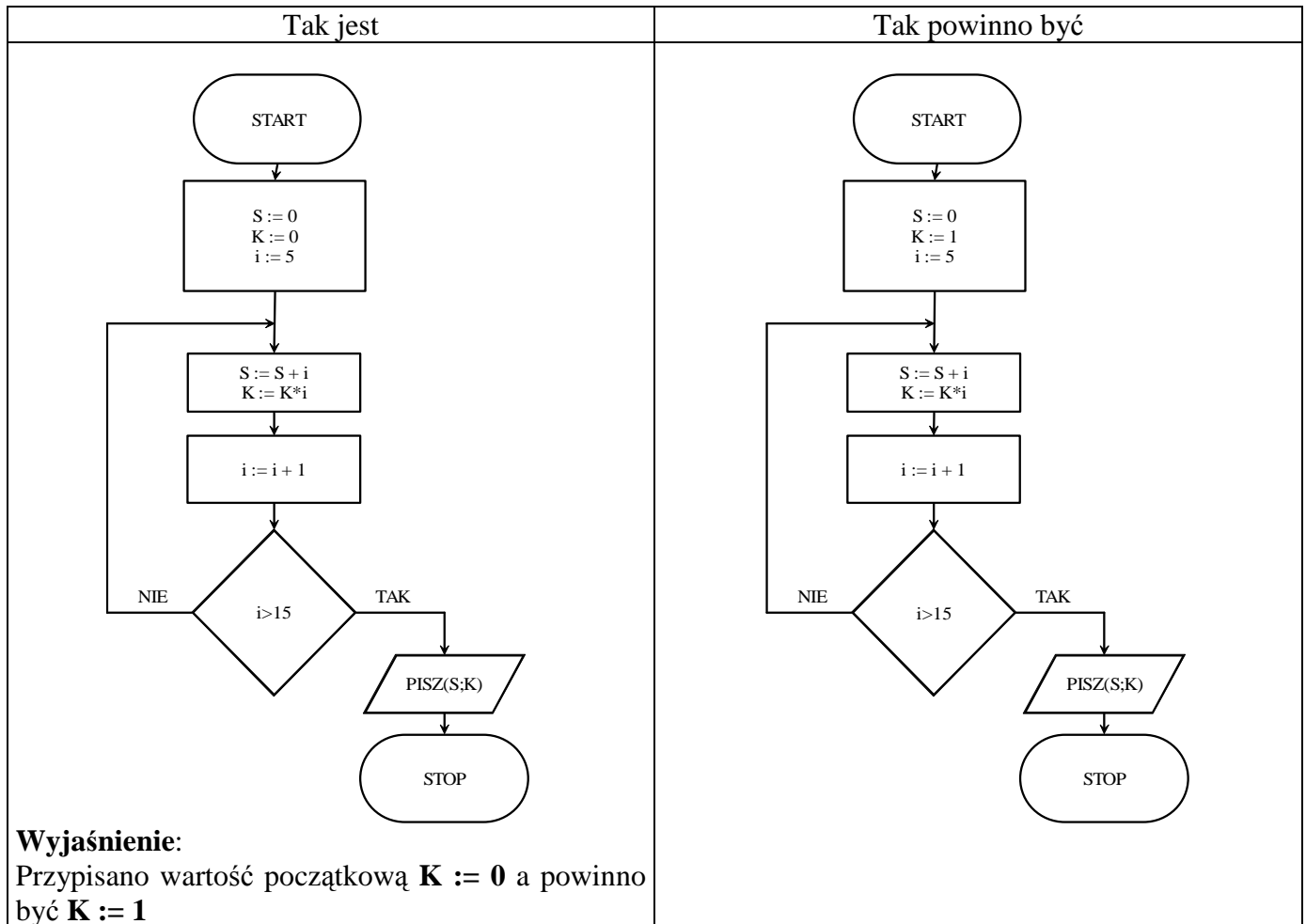
Przykładowa treść zadania:

Napisz algorytm za pomocą schematów blokowych, który oblicza iloczyn 5 elementów ciągu arytmetycznego: 1, 2, 3, 4, 5. Wynik wyświetl na ekranie monitora. Zrealizuj zadanie za pomocą instrukcji POWTARZAJ.

Zadanie nr 20

Obok przedstawiono schemat blokowy. Wskaż błąd w tym schemacie oraz ułóż do niego treść zadania.

Odpowiedź



Przykładowa treść zadania:

Napisz algorytm za pomocą schematów blokowych, który oblicza sumę oraz iloczyn 11 elementów ciągu arytmetycznego: 5, 6, Wynik wyświetl na ekranie monitora. Zrealizuj zadanie za pomocą instrukcji POWTARZAJ.

Zadanie nr 21

Napisz algorytm za pomocą schematy blokowego i pseudojęzyka, który oblicza sumę parzystych i iloczyn nieparzystych elementów ciągu arytmetycznego: 10,11 ...35.

Odpowiedź

| pseudojęzyk | schemat blokowy |
|--|---|
| <pre> START i := 10 s := 0 p := 1 DOPÓKI i <= 35 WYKONUJ POCZĄTEK JEŻELI (i MOD 2 = 0) TO s := s + i WPR p := p * i i := i + 1 KONIEC PISZ(s;p) STOP </pre> | <pre> graph TD Start([START]) --> Init[i := 1 s := 0 p := 1] Init --> Cond1{i <= 35} Cond1 -- TAK --> Cond2{i MOD 2 = 0} Cond1 -- NIE --> Print[/PISZ(s;p)/] Print --> Stop([STOP]) Cond2 -- TAK --> Sum[s := s + i] Cond2 -- NIE --> Prod[p := p * i] Sum --> Inc[i := i + 1] Prod --> Inc Inc --> Cond1 </pre> |

START

$i := 10$

$s := 0$

$p := 1$

POWTARZAJ

JEŻELI ($i \text{ MOD } 2 = 0$) TO

$s := s + i$

WPR

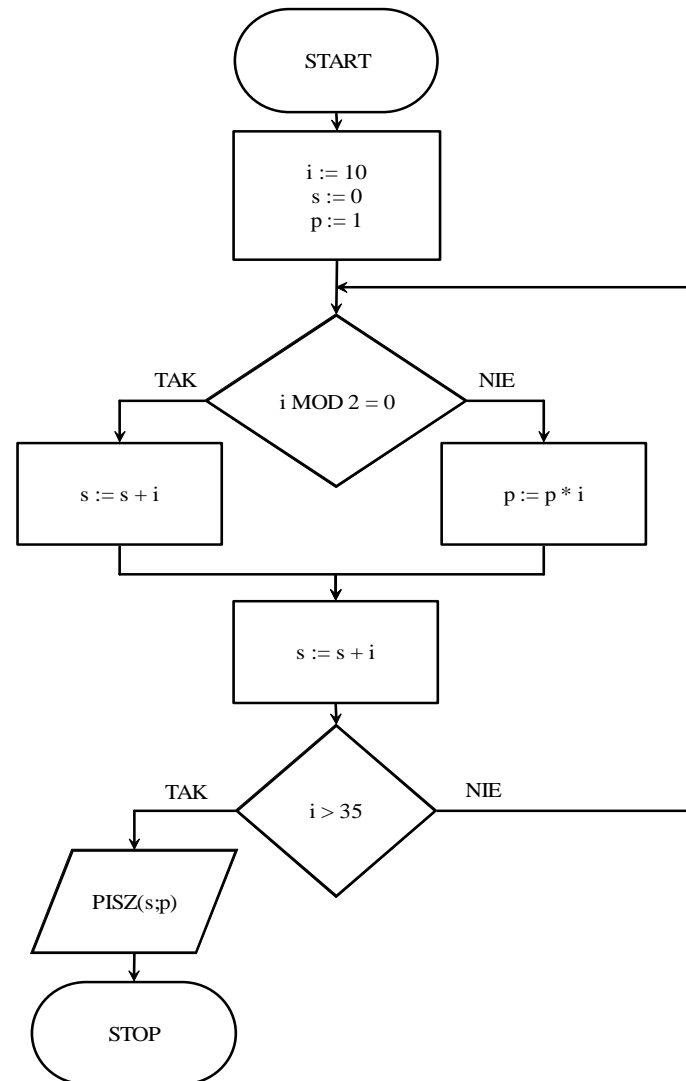
$p := p * i$

$i := i + 1$

AŻ $i > 35$

PISZ($s;p$)

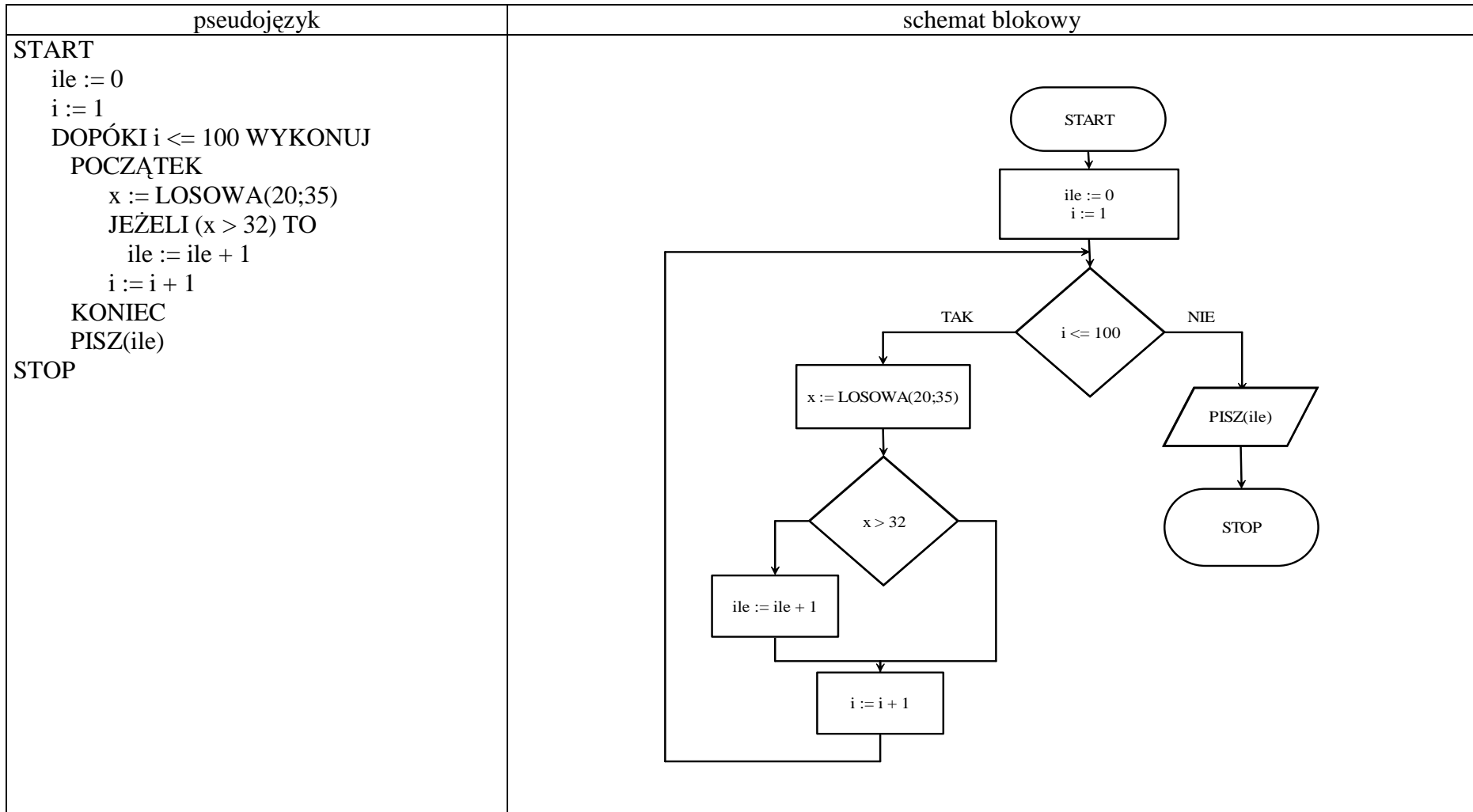
STOP



Zadanie nr 22

Napisz algorytm za pomocą schematu blokowego i pseudojęzyka, który generuje 100 liczb całkowitych z przedziału $\langle 20;35 \rangle$. Algorytm ma obliczać ile wygenerowano liczb większych od 32.

Odpowiedź



START

ile := 0

i := 1

POWTARZAJ

 x := LOSOWA(20;35)

 JEŻELI (x > 32) TO

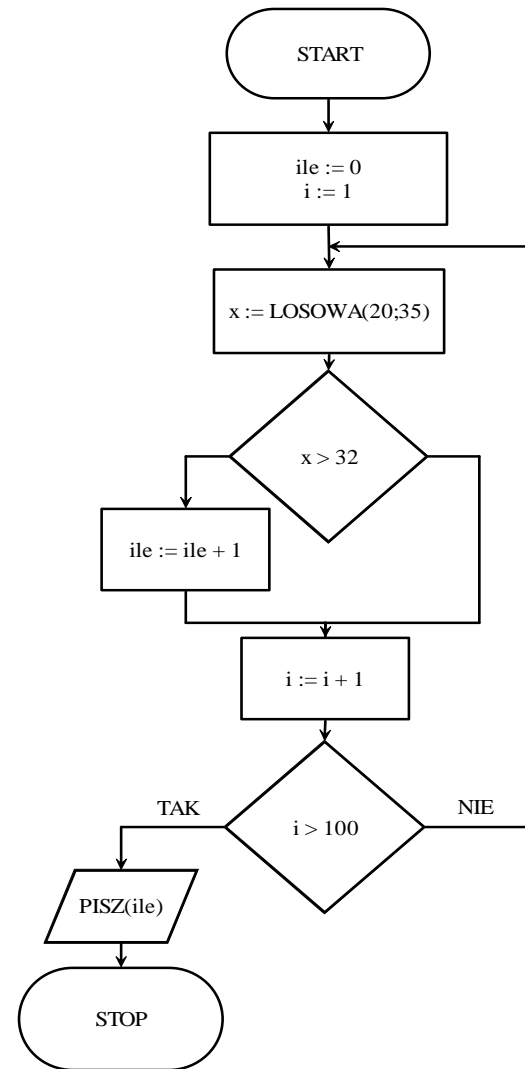
 ile := ile + 1

 i := i + 1

AŻ i > 100

PISZ(ile)

STOP



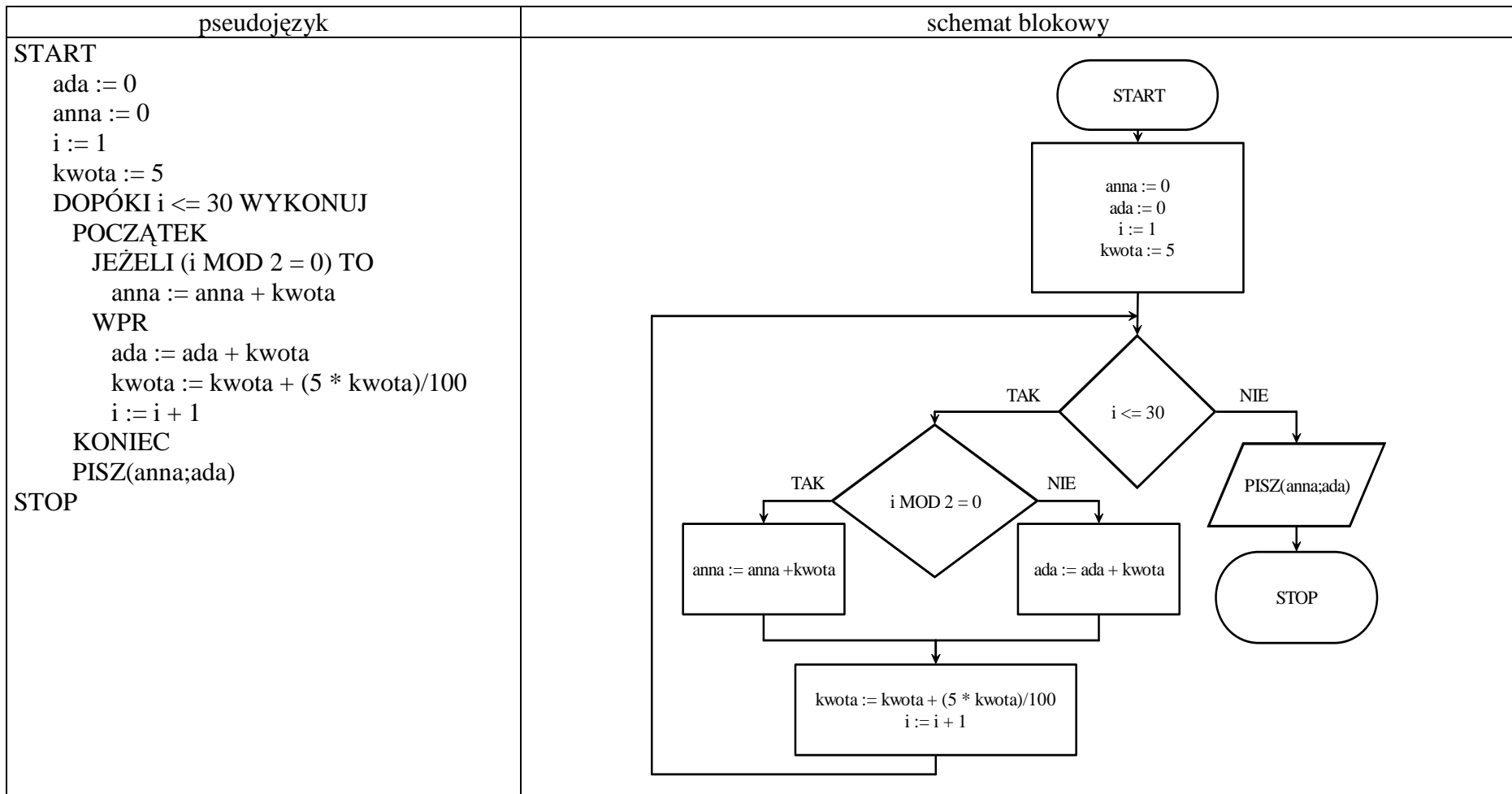
Zadanie nr 23

Anna i Ada otrzymują od taty pieniądze przez 30 dni w następujący sposób:

- pierwsza otrzymuje pieniądze Ada i otrzymała od taty pierwszego dnia 5 złotych
- każdego dnia pieniądze może otrzymać tylko jedna dziewczynka
- dziewczynki otrzymują pieniądze na przemian
- wysokość otrzymywanych kwot wzrasta codziennie o 5%

Napisz algorytm który obliczy ile pieniędzy zaoszczędzi Anna i Ada?

Odpowiedź



START

ada := 0

anna := 0

i := 1

kwota := 5

POWTARZAJ

JEŻELI (i MOD 2 = 0) TO

anna := anna + kwota

WPR

ada := ada + kwota

kwota := kwota + (5 * kwota)/100

i := i + 1

AŻ i > 30

PISZ(anna;ada)

STOP

