

UZUPEŁNIA ZDAJĄCY

KOD

--	--	--

PESEL

--	--	--	--	--	--	--	--	--	--	--	--

*miejsce
na naklejkę*

**EGZAMIN MATURALNY
Z INFORMATYKI**

POZIOM ROZSZERZONY

CZĘŚĆ I

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 9 stron. Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. Rozwiązania i odpowiedzi zamieść w miejscu na to przeznaczonym.
3. Pisz czytelnie. Używaj długopisu/pióra tylko z czarnym tuszem/atramentem.
4. Nie używaj korektora, a błędne zapisy wyraźnie przekreśl.
5. Pamiętaj, że zapisy w brudnopisie nie podlegają ocenie.
6. Wpisz obok zadeklarowane (wybrane) przez Ciebie na egzamin środowisko komputerowe, kompilator języka programowania oraz program użytkowy.
7. Jeżeli rozwiązaniem zadania lub jego części jest algorytm, to zapisz go w wybranej przez siebie notacji: listy kroków, schematu blokowego lub języka programowania, który wybierasz na egzamin.
8. Na tej stronie oraz na karcie odpowiedzi wpisz swój numer PESEL i przyklej naklejkę z kodem.
9. Nie wpisuj żadnych znaków w części przeznaczonej dla egzaminatora.



10 maja 2017

**Godzina rozpoczęcia:
14:00**

WYBRANE:

.....
(środowisko)
.....
(kompilator)
.....
(program użytkowy)

**Czas pracy:
90 minut**

**Liczba punktów
do uzyskania: 20**

MIN-R1_1P-172

Rozważmy problem sortowania ciągu liczb całkowitych z przedziału $[1..k]$ dla znanej całkowitej wartości k . Poniżej prezentujemy algorytm rozwiązujący ten problem, zgodny z następującą specyfikacją:

<i>Dane:</i>	n, k	– liczby całkowite dodatnie
	$T[1..n]$	– ciąg liczb całkowitych z zakresu $[1..k]$
<i>Wynik:</i>	$W[1..n]$	– uporządkowany niemalejąco ciąg liczb z tablicy $T[1..n]$

```

dla  $i=1..k$  wykonuj
     $Liczba\_wystapien[i] \leftarrow 0$ 
dla  $i=1..n$  wykonuj
     $Liczba\_wystapien[T[i]] \leftarrow Liczba\_wystapien[T[i]] + 1$ 
 $p \leftarrow 1$ 
dla  $j=1..k$  wykonuj
    dla  $i=1..Liczba\_wystapien[j]$  wykonuj
         $W[p] \leftarrow j$ 
         $p \leftarrow p+1$ 

```

Uzupełnij poniższą tabelę – podaj końcową zawartość tablicy *Liczba_wystapien* dla odpowiednich danych wejściowych.

n	k	$T[1..n]$	Końcowa zawartość $Liczba_wystapien[1..k]$
10	5	[1, 2, 3, 4, 5, 1, 2, 3, 4, 4]	[2, 2, 2, 3, 1]
5	10	[1, 3, 3, 5, 10]	
5	5	[5, 5, 5, 5, 5]	
10	4	[1, 2, 3, 1, 2, 3, 1, 1, 2, 3]	

[illegible]

A full-page sheet of white graph paper with a light gray grid. The grid consists of small squares, approximately 10 units wide by 10 units high. There are no margins or additional markings on the page.

Strona 4 z 9

Zadanie 2. Ciąg Pentanacciego

Rozważmy ciąg liczb $p_0, p_1, p_2, \dots$ zdefiniowany w następujący sposób:

$$\begin{cases} p_0 = 0 \\ p_1 = 1 \\ p_2 = 1 \\ p_3 = 2 \\ p_4 = 4 \\ p_n = p_{n-1} + p_{n-2} + p_{n-3} + p_{n-4} + p_{n-5} \text{ dla } n \geq 5 \end{cases}$$

Zadanie 2.1 (0–2)

Uzupełnij poniższą tabelę.

n	p_n
5	8
7	
9	

Zadanie 2.2 (0–3)

Poniżej prezentujemy algorytm, który powinien wyznaczać n -ty element podanego ciągu. Uzupełnij luki w algorytmie tak, aby jego działanie było zgodne z podaną specyfikacją.

Specyfikacja:

Dane: n – nieujemna liczba całkowita

Wynik: w – liczba całkowita równa p_n

Algorytm:

$tab[0] \leftarrow 0$

$tab[1] \leftarrow 1$

$tab[2] \leftarrow 1$

$tab[3] \leftarrow 2$

$tab[4] \leftarrow 4$

$i \leftarrow 5$

dopóki $i \leq \dots$ **wykonuj**

$temp \leftarrow tab[0] + tab[1] + tab[2] + tab[3] + tab[4]$

$tab[\dots \bmod 5] \leftarrow temp$

$i \leftarrow i + 1$

$w \leftarrow \dots$

Uwaga: $a \bmod b$ oznacza resztę z dzielenia liczby a przez liczbę b .

A large grid of graph paper with 20 columns and 10 rows. The grid is composed of small squares, with a slightly larger square at the top left corner, likely for a title or header. The grid is empty and ready for use.

$$\left\{ \begin{array}{l} r_0 = 0 \\ r_1 = 1 \\ r_2 = 1 \\ r_3 = 0 \\ r_4 = 0 \\ r_n = (r_{n-1} + r_{n-2} + r_{n-3} + r_{n-4} + r_{n-5}) \bmod 2 \text{ dla } n \geq 5 \end{array} \right.$$
[illegible]

Wypełnia egzaminator	Nr zadania	2.1.	2.2.	2.3.
	Maks. liczba pkt.	2	3	3
	Uzyskana liczba pkt.			

Zadanie 3. Test

W każdym z poniższych zadań oceń, które z podanych zdań są prawdziwe. Zaznacz P, jeśli zdanie jest prawdziwe, lub F – jeśli jest fałszywe.

Zadanie 3.1 (0–1)

Dane są tablica $A[1..6]$ o zawartości $[6, 2, -1, 5, 1, 2]$ oraz następujący fragment algorytmu:

```
s ← 0
n ← 3
i ← 6
dopóki i > n - 1 wykonuj
    s ← s + A[i]
    i ← i - 1
```

Po wykonaniu tego algorytmu spełniony jest warunek

s jest parzyste.	P	F
$s = 7$.	P	F
$s > 6$.	P	F
$s = 3$.	P	F

Zadanie 3.2 (0–1)

Realizacji usług poczty elektronicznej służy protokół

SMTP.	P	F
IMAP.	P	F
EMAIL.	P	F
POP3.	P	F

Zadanie 3.3 (0–1)

Liczbą większą od $150_{(10)}$ jest

$10011001_{(2)}$	P	F
$1222_{(4)}$	P	F
$227_{(8)}$	P	F
$9B_{(16)}$	P	F

Zadanie 3.4 (0–1)

Obrazy rastrowe

są reprezentowane jako tablice pikseli, co powoduje istotną utratę jakości przy powiększaniu obrazu.	P	F
tworzone są przy użyciu wyrażeń matematycznych opisujących występujące w obrazie odcinki, krzywe, elipsy itp.	P	F
mogą być wprowadzane do komputera przy użyciu urządzeń takich jak aparat cyfrowy lub skaner.	P	F
mogą powstać w efekcie cyfrowego zapisu obrazu widzialnego.	P	F

Zadanie 3.5 (0–1)

Algorytm zwany sitem Eratostenesa opierający się na „wykreślaniu” wielokrotności kolejnych (niewykreślonych wcześniej) liczb naturalnych służy wyznaczeniu

największego wspólnego dzielnika dwóch liczb.	P	F
najmniejszej wspólnej wielokrotności dwóch liczb.	P	F
liczb pierwszych z zadanego przedziału.	P	F
potęg dwójki z zadanego przedziału.	P	F

Zadanie 3.6 (0–1)

Przykładem programu, który służy do tłumaczenia instrukcji kodu źródłowego **programu komputerowego** na język maszynowy, jest

walidator.	P	F
kompilator.	P	F
edytor tekstu.	P	F
defragmentator.	P	F

Wypełnia egzaminator	Nr zadania	3.1.	3.2.	3.3.	3.4.	3.5.	3.6.
	Maks. liczba pkt.	1	1	1	1	1	1
	Uzyskana liczba pkt.						

BRUDNOPIS (*nie podlega ocenie*)