ARKUSZ ZAWIERA INFORMACJE PRAWNIE CHRONIONE DO MOMENTU ROZPOCZĘCIA EGZAMINU!

Miejsce na naklejkę

MIN-R1_1P-082

EGZAMIN MATURALNY Z INFORMATYKI

MAJ ROK 2008

POZIOM ROZSZERZONY

CZĘŚĆ I

Czas pracy 90 minut

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 13 stron (zadania 1 – 3). 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. Na karcie odpowiedzi wpisz swoją datę urodzenia i PESEL. Nie wpisuj żadnych znaków w części przeznaczonej dla egzaminatora.

Życzymy powodzenia!

Za rozwiązanie wszystkich zadań można otrzymać łącznie **40 punktów**

Wypełnia zdający przed rozpoczęciem pracy

PESEL ZDAJĄCEGO

KOD ZDAJĄCEGO

Zadanie 1. Potęgi (*14 pkt*)

W poniższej tabelce podane są wartości kolejnych potęg liczby 2:

k	0	1	2	3	4	5	6	7	8	9	10
2^k	1	2	4	8	16	32	64	128	256	512	1024

Ciąg $a=(a_0, a_1, a_2,...)$ definiujemy następująco:

a_k = reszta z dzielenia liczby 2^k przez 10 dla $k = 0, 1, 2, ...$

a) Korzystając z definicji, podaj 16 pierwszych wyrazów ciągu *a*. Wyniki umieść w poniższej tabelce:

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a_k	*1*	*2*	*4*	*8*	*6*	*2*	*4*	*8*	*6*	*2*	*4*	*8*	*6*	*2*	*4*	*8*

Uwaga: w dalszej części tego zadania możesz przyjąć, że operacje arytmetyczne na liczbach całkowitych (dodawanie, odejmowanie, mnożenie, dzielenie całkowite, reszta z dzielenia) wykonywane są w czasie stałym, niezależnie od wielkości argumentów.

b) W wybranej przez siebie notacji (lista kroków, schemat blokowy lub język programowania) podaj algorytm, który dla danej nieujemnej liczby całkowitej k wyznacza resztę z dzielenia liczby 2^k przez 10. Np. dla $k = 15$ wynikiem działania Twojego algorytmu powinno być 8.

Przy ocenie Twojego rozwiązania będzie brana pod uwagę zarówno poprawność zaproponowanego algorytmu, jak i jego złożoność czasowa, czyli liczba operacji arytmetycznych wykonywanych w trakcie obliczania wyniku.

Specyfikacja:

Dane: Liczba całkowita $k \geq 0$.

Wynik: Reszta z dzielenia 2^k przez 10.

Algorytm

krok 1: jeżeli k = 0, to wynikiem jest 1

krok 2: w przeciwnym przypadku

krok 2.1: policz resztę z dzielenia k przez 4

krok 2.2: jeżeli reszta = 0, to wynikiem jest 6

krok 2.3: jeżeli reszta = 1, to wynikiem jest 2

krok 2.4: jeżeli reszta = 2, to wynikiem jest 4

krok 2.5: jeżeli reszta = 3, to wynikiem jest 8

c) Podaj w wybranej przez siebie notacji (lista kroków, schemat blokowy lub język programowania) algorytm obliczania liczby a^n, gdy a jest liczbą całkowitą, natomiast n jest potęgą liczby 2 ($n = 2^k$ dla pewnej liczby całkowitej $k \geq 0$). Przy ocenie Twojego rozwiązania będzie brana pod uwagę złożoność czasowa (w zależności jedynie od n) zaproponowanego algorytmu, czyli liczba operacji arytmetycznych wykonywanych w trakcie obliczania wyniku.

Wskazówka: zauważ, że $a^n = a^{\frac{n}{2}} \cdot a^{\frac{n}{2}}$, dla n>1.

Specyfikacja:

Dane: Liczby całkowite a i n, gdzie $n = 2^k$ dla pewnej liczby całkowitej $k \geq 0$.

Wynik: Liczba a^n.

Algorytm

krok 1: p := a

krok 2: dopóki n > 1 wykonuj

*krok 2.1: p := p * p*

krok 2.2: n := n div 2

krok 3: wynikiem jest p

<table>
<tr><td rowspan="3">Wypełnia egzaminator!</td><td>Nr zadania</td><td>1 a)</td><td>1 b)</td><td>1 c)</td></tr>
<tr><td>Maks. liczba pkt</td><td>2</td><td>5</td><td>7</td></tr>
<tr><td>Uzyskana liczba pkt</td><td></td><td></td><td></td></tr>
</table>

Zadanie 2. Słowa (*14 pkt*)

Niech $A = \{a, b\}$ będzie dwuliterowym alfabetem. Napisem nad alfabetem A nazywamy skończony ciąg znaków z tego alfabetu o długości większej od zera. Np. takimi napisami są:

a, *ab*, *aba*, *baba*, *aaaa*

Długość napisu w będziemy oznaczać przez $|w|$. Zatem $|aba| = 3$.

Jeżeli w_1 i w_2 są napisami, to przez w_1w_2 będziemy oznaczali napis zbudowany z napisu w_1 i z następującego po nim napisu w_2. Np. dla $w_1 = ab$ i $w_2 = aa$, $w_1w_2 = abaa$.

Zdefiniujemy teraz napisy 2-regularne. Każdy napis złożony tylko z jednej litery jest 2-regularny. Jeżeli napis w jest 2-regularny, to napis ww jest też 2-regularny. Żadne inne napisy nie są 2-regularne.

Oto procedura rekurencyjna **2REG**(w), która sprawdza, czy dany napis w nad alfabetem A jest 2-regularny.

Specyfikacja:

Dane: napis w o długości n ($n \geq 1$), składający się z liter należących do alfabetu A.

Wynik: odpowiedź *TAK*, jeśli napis w jest napisem 2-regularnym; odpowiedź *NIE*, jeśli napis w nie jest napisem 2-regularnym.

2REG(w);

krok 1: jeśli $|w| = 1$, to wynikiem jest *TAK*

krok 2: jeśli $|w| > 1$ i $|w|$ jest nieparzyste, to wynikiem jest *NIE*

krok 3: jeśli $|w| > 1$ i $|w|$ jest parzyste, to:

krok 3.1: podziel napis w na dwa napisy w_1 i w_2 o takiej samej długości i takie, że $w = w_1w_2$

krok 3.2: jeśli $w_1 \neq w_2$, to wynikiem jest *NIE*

krok 3.3: wynikiem jest wynik wywołania **2REG**(w_1)

a) Wypisz parametry wszystkich wywołań rekurencyjnych funkcji **2REG** dla poniższych napisów oraz podaj wynik jej działania:
   i. *aabbaabb*
   ii. *aaaaaaaa*
   iii. *bbbbbbbbbbbbbbbbbb*

np.: dla napisu w = *abab*, parametry wszystkich wywołań rekurencyjnych funkcji **2REG** i wynik jej działania są następujące:

abab→*ab*→*NIE*

aabbaabb $\rightarrow$ *aabb* $\rightarrow$ *NIE*

aaaaaaaa $\rightarrow$ *aaaa* $\rightarrow$ *aa* $\rightarrow$ *a* $\rightarrow$ *TAK*

bbbbbbbbbbbbbbbbbb $\rightarrow$ *bbbbbbbbb* $\rightarrow$ *bbbbb* $\rightarrow$ *NIE*

b) Jakiej długości są napisy 2-regularne? Odpowiedź uzasadnij.

Długość napisu musi być potęgą liczby 2, gdyż napis jednoliterowy jest 2-regularny, a każdy napis 2-regularny o długości większej od 1 powstaje z połączenia dwóch napisów 2-regularnych o takiej samej długości, a zatem ma długość dwa razy większą od długości każdego z tych napisów.

c) Ile jest napisów 2-regularnych o długości n ($n \geq 1$) nad alfabetem A? Odpowiedź uzasadnij.

Są tylko dwa napisy 2-regularne o długości n, gdy n jest potęgą liczby 2. Jeden napis to napis składający się tylko z liter ***a****, drugi napis to napis składający się tylko z liter* ***b****. Jeśli n nie jest potęgą liczby 2, to nie ma napisów 2-regularnych o tej długości.*

Jednoliterowy napis 2-regularny składa się albo z litery a, albo z litery b. Każdy napis 2-regularny o długości > 1 powstaje z połączenia dwóch identycznych, zbudowanych z tej samej litery, napisów 2-regularnych.

d) Pewnym uogólnieniem napisów 2-regularnych są napisy 3-regularne.
Każdy napis jednoliterowy jest 3-regularny. Jeśli napis w jest 3-regularny, to każdy z napisów wxw, wwx, gdzie x jest dowolnym napisem nad alfabetem A i takim, że długość x jest taka sama jak długość w, jest napisem 3-regularnym. Żaden inny napis nie jest 3-regularny.
Przykładowymi napisami 3-regularnymi są: *a*, *aba*, *abaabaaaa*.
Ale *aaaabaaba* nie jest 3-regularny.

Napisz w wybranej przez siebie notacji (lista kroków, schemat blokowy lub język programowania) algorytm zgodny ze specyfikacją, który sprawdza 3-regularność danego napisu.

Specyfikacja:
Dane: napis w, o długości n ($n \geq 1$), składający się z liter należących do alfabetu A.

Wynik: odpowiedź *TAK*, jeśli napis w jest napisem 3-regularnym; odpowiedź *NIE*, jeśli napis w nie jest napisem 3-regularnym.

Algorytm

3REG(w)

krok 1: n := długość słowa w

krok 2: jeżeli n = 1, to wynikiem jest TAK

krok 3: jeżeli n > 1 i n nie jest podzielne przez 3, to wynikiem jest NIE

krok 4: jeżeli n jest podzielne przez 3, to:

krok 4.1: podziel słowo w na 3 podsłowa w_1, w_2, w_3 o równych długościach i takie, że $w = w_1 w_2 w_3$

krok 4.2: jeżeli (w_1=w_2) lub (w_1=w_3), to wynikiem jest wynik wywołania 3REG(w_1)

krok 4.3: w przeciwnym razie wynikiem jest NIE

Wypełnia egzaminator!	Nr zadania	2 a)	2 b)	2 c)	2 d)
	Maks. liczba pkt	3	2	2	7
	Uzyskana liczba pkt				

Zadanie 3. Test (*12 pkt*)

Podpunkty a) – l) zawierają po trzy odpowiedzi, z których każda jest albo prawdziwa, albo fałszywa. Zdecyduj, które z podanych odpowiedzi są prawdziwe (**P**), a które fałszywe (**F**). **Zaznacz znakiem X** odpowiednią rubrykę w tabeli.

a) Dla poniższego algorytmu dane stanowi skończony ciąg liczbowy zawierający co najmniej jedną liczbę:

1. *i* := 0
2. *wynik* := 0
3. dopóki nie przetworzono wszystkich liczb w ciągu wykonuj:
   i. *x* := kolejna liczba
   ii. *wynik* := (*i***wynik*+*x*)/(*i*+1)
   iii. *i* := *i*+1
4. wypisz wynik

Uwaga: „:=” oznacza instrukcję przypisania.

Wynikiem działania tego algorytmu jest

	P	F
suma podanych liczb.		X
średnia arytmetyczna podanych liczb.	X	
średnia geometryczna podanych liczb.		X

b) Poszukując numeru telefonu w książce telefonicznej wiele osób korzysta z następującego algorytmu: otwieramy książkę mniej więcej w połowie. Jeśli szukane nazwisko w kolejności alfabetycznej jest wcześniej niż nazwisko, na które trafiliśmy, otwieramy książkę w połowie, licząc od początku do miejsca, w którym się znajdujemy. W przeciwnym przypadku bierzemy pod uwagę drugą połowę książki. Postępujemy podobnie dla tej części książki, którą wybraliśmy, aż do momentu, kiedy jesteśmy blisko szukanego nazwiska. Wtedy wystarczy już przejrzeć kilka stron. Ten sposób postępowania jest zastosowaniem w praktyce strategii

	P	F
dziel i zwyciężaj.	X	
zachłannej.		X
porządkowania ciągu elementów.		X

c) Urządzenie, które pobiera dane cyfrowe z komputera i zamienia je na sygnały analogowe przesyłane w sieci telefonicznej to

	P	F
karta sieciowa.		X
router.		X
modem.	X	

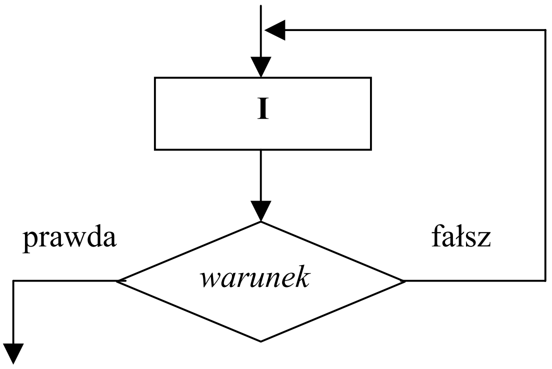
d) Zapis $1010_{(p)}$ oznacza, że 1010 jest zapisem pewnej liczby w systemie pozycyjnym o podstawie *p*. Zaznacz, która z poniższych równości jest prawdziwa:

	P	F
$1010_{(2)} = 10_{(10)}$	X	
$12_{(10)} = 1110_{(2)}$		X
$67_{(10)} = 1000011_{(2)}$	X	

e) Kod ASCII znaku zero wynosi 48, a kodem małej litery „a” jest 97.

	P	F
Kodem znaku „3” jest liczba $00110100_{(2)}$.		X
Kodem znaku „4” jest liczba $01100000_{(2)}$.		X
Kodem małej litery „f” jest liczba $01100110_{(2)}$.	X	

f) Poniższy schemat blokowy opisuje instrukcję powtarzania, w której



	P	F
liczba powtórzeń instrukcji **I** nie zależy od warunku *warunek*.		X
instrukcja **I** jest wykonywana co najmniej raz.	X	
jeśli *warunek* **nie jest** spełniony, to następuje zakończenie powtarzania.		X

g) Do szyfrowania informacji służy

	P	F
algorytm RSA.	X	
algorytm Euklidesa.		X
algorytm Hornera.		X

h) Adresy IP składają się z czterech liczb z zakresu od 0 do 255, które zapisuje się oddzielone kropkami, np. 130.11.121.94. Pierwsza z liczb zapisana binarnie na ośmiu bitach pozwala określić, do jakiej klasy należy adres. Adresy klasy B mają na dwóch pierwszych bitach (licząc od lewej strony) wartości odpowiednio 1 i 0. Adresy klasy C mają na pierwszych trzech pozycjach wartości 1, 1 i 0.

	P	F
Adres 128.12.67.90 należy do klasy B.	X	
Adres 191.12.56.1 należy do klasy C.		X
Adres 192.14.56.10 należy do klasy B.		X

i) Skrótem nazwy protokołu sieciowego jest

	P	F
FTP.	X	
SSH.	X	
OSI.		X

j) Plik graficzny zawiera obrazek o rozmiarach 1024 na 768 pikseli zapisany z użyciem 256 kolorów. Do zapisania tego pliku (bez użycia kompresji) potrzebne jest

	P	F
786432 bitów.		X
786432 bajtów.	X	
786432 kilobajtów.		X

k) Nazwą nośnika pamięci zewnętrznej jest

	P	F
płyta CD.	X	
pamięć flash.	X	
pamięć cache.		X

l) Asymetryczne metody szyfrowania wymagają

	P	F
używania takich samych kluczy do szyfrowania i deszyfrowania wiadomości.		X
używania różnych kluczy do szyfrowania i deszyfrowania wiadomości.	X	
ujawniania klucza służącego do szyfrowania.	X	

Wypełnia egzaminator!	Nr zadania	3 a)	3 b)	3 c)	3 d)	3 e)	3 f)	3 g)	3 h)	3 i)	3 j)	3 k)	3 l)
	Maks. liczba pkt	1	1	1	1	1	1	1	1	1	1	1	1
	Uzyskana liczba pkt												

BRUDNOPIS