Miejsce
na naklejkę

MIN-R1_1P-092

EGZAMIN MATURALNY
Z INFORMATYKI

POZIOM ROZSZERZONY

CZĘŚĆ I

Czas pracy 90 minut

MAJ
ROK 2009

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 9 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. Wpisz obok 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 wybrałeś/aś na egzamin.
8. Na karcie odpowiedzi wpisz swoją datę urodzenia i PESEL. Nie wpisuj żadnych znaków w części przeznaczonej dla egzaminatora.

Życzymy powodzenia!

WYBRANE:

....
(środowisko)

....
(kompilator)

....
(program użytkowy)

Za rozwiązanie wszystkich zadań można otrzymać łącznie **30 punktów**

Wypełnia zdający przed rozpoczęciem pracy

PESEL ZDAJĄCEGO

KOD ZDAJĄCEGO

Zadanie 1. Test *(6 pkt)*

Zaznacz znakiem X w odpowiedniej kolumnie P lub F, która odpowiedź jest prawdziwa, a która fałszywa.

a) Przeanalizuj poniższy algorytm (:= oznacza instrukcję przypisania)

1. *m*:=0
2. *n*:=6
3. jeśli *m*>*n* to wykonaj krok 7.
4. *m*:=*m*+1
5. pisz *m*
6. przejdź do kroku 3.
7. stop

	P	F
Wykonywanie algorytmu zakończy się po wypisaniu liczb od 1 do 7.		
Po pierwszym sprawdzeniu warunku w kroku 3. nie zostaną wykonane kroki: 4., 5., 6. i wykonywanie algorytmu zakończy się.		
Wykonywanie algorytmu zakończy się po wypisaniu liczb od 0 do 6.		
Sprawdzenie warunku $m > n$ wykonane zostanie dokładnie 8 razy.		

b) 434 176 bity to

	P	F
53 kB.		
53 MB.		
mniej niż 50 kB.		
54 272 bajty.		

c) Liczba dziesiętna 83 jest reprezentowana przez

	P	F
$(63)_{16}$		
$(121)_8$		
$(1103)_4$		
$(10100011)_2$		

d) 8-bitowa reprezentacja pewnej liczby dziesiętnej zapisanej w kodzie U2 ma postać **11111110**. Tą liczbą jest

	P	F
–2.		
–126.		
–1.		
254.		

e) Schemat Hornera znajduje zastosowanie przy

	P	F
obliczaniu pola powierzchni figur płaskich.		
obliczaniu wartości wielomianu przy minimalnej liczbie operacji mnożenia.		
szybkim sortowaniu dużych zbiorów danych.		
znajdowaniu najmniejszego elementu w zbiorze.		

Wypełnia egzaminator!	**Nr zadania**	**1 a)**	**1 b)**	**1 c)**	**1 d)**	**1 e)**
	Maks. liczba pkt	**2**	**1**	**1**	**1**	**1**
	Uzyskana liczba pkt					

Zadanie 2. Punkty kratowe (*14 pkt*)

Punkt kratowy to punkt, którego współrzędne w układzie kartezjańskim są liczbami całkowitymi.

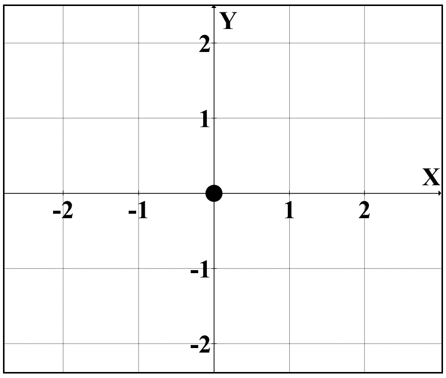
Przykłady punktów kratowych:

$(-100,101)$, $(1,1)$, $(0,0)$, $(-1,-3)$.

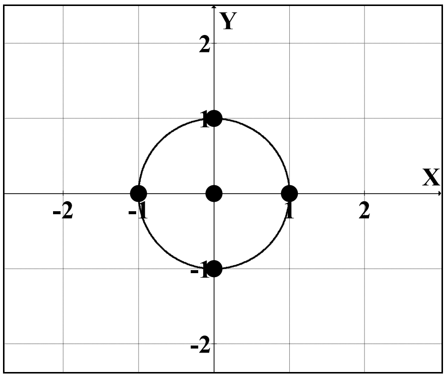
Rozważamy koła o środku w początku układu współrzędnych. Dla nieujemnej liczby rzeczywistej R przez $K(R)$ oznaczmy koło o promieniu R (brzeg koła należy do koła). Niech $N(R)$ będzie liczbą punktów kratowych zawartych w kole $K(R)$.

Przykłady:

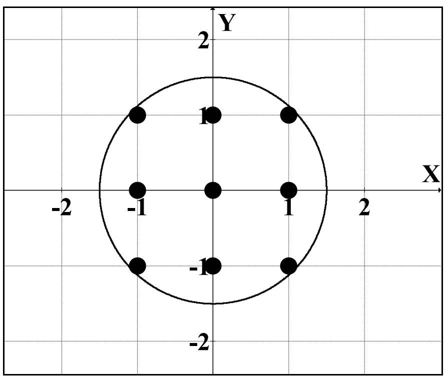
Jeżeli $R = 0$, to $N(R) = 1$.



Jeżeli $R = 1$, to w kole $K(R)$ mieści się pięć punktów kratowych, czyli $N(R) = 5$.



Jeżeli $R = 1{,}5$, to w kole $K(R)$ mieści się dziewięć punktów kratowych, zatem $N(R) = 9$.



a) Uzupełnij poniższą tabelę:

Promień koła R	*Liczba punktów kratowych* $N(R)$
2,01	
4,50	

b) Zaproponuj algorytm zapisany w wybranej przez siebie notacji (lista kroków, schemat blokowy lub język programowania, który wybrałeś/aś na egzamin) obliczający liczbę punktów kratowych zawierających się w kole o promieniu *R*.

Specyfikacja:

Dane: *R* – promień koła o środku znajdującym się w początku układu współrzędnych $(0,0)$; **liczba całkowita nieujemna**.

Wynik: liczba całkowita $N(R)$ – liczba punktów kratowych zawierających się w kole o środku $(0,0)$ i promieniu *R*

Algorytm:

<table>
<tr><td rowspan="3">Wypełnia egzaminator!</td><td>Nr zadania</td><td>2 a)</td><td>2 b)</td></tr>
<tr><td>Maksymalna liczba pkt.</td><td>4</td><td>10</td></tr>
<tr><td>Uzyskana liczba pkt</td><td></td><td></td></tr>
</table>

Zadanie 3. Największy wspólny dzielnik – NWD *(10 pkt)*

Algorytm opisany w Księdze VII *Elementów* Euklidesa pozwala szybko obliczyć największy wspólny dzielnik dwóch liczb naturalnych *a* i *b* – $nwd(a,b)$, z których co najmniej jedna jest większa od 0. Oto rekurencyjny sposób obliczania $nwd(a,b)$:

$$nwd(a,b) = \begin{cases} a \text{ dla } b = 0 \\ nwd(\,b, a \bmod b\,) \text{ dla } b \geq 1 \end{cases}$$

gdzie: mod – operator dzielenia modulo; wynikiem jego działania jest **reszta** z dzielenia *a* przez *b*, na przykład 19 mod 7 = 5.

Przykład: $nwd(16,12) = nwd(12,4) = nwd(4,0) = 4$ – funkcja *nwd* jest wywoływana w tym przypadku 3 razy:

a	b	reszta = a mod b	wywołanie
16	**12**	4	(1)
12	4	0	(2)
4 (wynik)	0	-	(3)

a) Podaj liczbę wywołań funkcji dla $a = 56$ **i** $b = 72$ oraz dla $a = 72$ **i** $b = 56$.

b) Podaj w wybranej przez siebie notacji (lista kroków, schemat blokowy lub język programowania, który wybrałeś/aś na egzamin) **nierekurencyjny** algorytm obliczania wartości funkcji $nwd(a,b)$ wraz ze specyfikacją.

Specyfikacja:

Dane: ....

....

Wynik: ....

....

Wypełnia egzaminator!	Nr zadania	3a)	3 b)
	Maksymalna liczba pkt	**2**	**8**
	Uzyskana liczba pkt		

BRUDNOPIS