|  |  |
| --- | --- |
| **WPISUJE ZESPÓŁ NADZORUJĄCY***miejsce**na naklejkę* |  |
|  |
|  **KOD PESEL** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |

|  |  |  |
| --- | --- | --- |
| **EGZAMIN MATURALNY Z INFORMATYKI****Poziom rozszerzony****Część I****Przykładowy arkusz egzaminacyjny**dla niewidomych (A6)Data: **18 grudnia 2014 r.**Czas pracy: **do 90 minut**Liczba punktów do uzyskania: **15**

|  |  |
| --- | --- |
| **WPISUJE ZESPÓŁ NADZORUJĄCY** | WYBRANE:**.................................................**(środowisko)**.................................................**(kompilator)**.................................................**(program użytkowy) |

 |
| **Instrukcja dla zdającego**1. Obok każdego numeru zadania podana jest maksymalna liczba punktów, którą można uzyskać za jego poprawne rozwiązanie.
2. Odpowiedzi zapisuj na kartkach dołączonych do arkusza, na których zespół nadzorujący wpisał Twój numer PESEL.
3. W razie pomyłki zapunktuj.
4. W czasie trwania egzaminu zdający może korzystać z kalkulatora prostego.
 |

Zadanie 1. Liczby Armstronga (0–5)

Liczba całkowita złożona z *n* cyfr jest liczbą Armstronga (narcystyczną) jeżeli jest sumą swoich cyfr podniesionych do potęgi *n*.
Na przykład: 153=13+53+33=1+125+27.

W tym zadaniu zajmiemy się przygotowaniem algorytmu sprawdzającego czy dana liczba jest liczbą Armstronga.

a) Sprawdź, czy liczby 6, 407, 2278 są liczbami Armstronga. (0–1)

b) W wybranej przez siebie notacji (schemat blokowy, lista kroków, wybrany przez Ciebie język programowania) zapisz algorytm zapisujący poszczególne cyfry liczby *k* w tablicy cyfry[] w kolejności od najmniej do najbardziej znaczącej oraz zwracający liczbę cyfr jej zapisu dziesiętnego.(0–2)

Specyfikacja:
*Dane:* *k* – liczba całkowita,
*Wynik:* *n* – liczba cyfr w zapisie dziesiętnym liczby *k*,
cyfry[] – tablica zawierająca kolejne cyfry zapisu dziesiętnego liczby *k*, w kolejności od najmniej znaczącej do najbardziej znaczącej.
*Przykład:*
*Dane:* *k* =54321
*Wynik:* *n* =5, cyfry[]=[1,2,3,4,5]

c) W wybranej przez siebie notacji (schemat blokowy, lista kroków, wybrany przez Ciebie język programowania) zapisz algorytm sprawdzający, czy zapisana w tablicy cyfry[] liczba *k* jest liczbą narcystyczną. W swoim algorytmie załóż, że posiadasz funkcję wyliczającą *n*-tą potęgę liczby *i*, oraz zapisane w tablicy cyfry[] kolejne cyfry zapisu dziesiętnego liczby *k*. (0-2)

Specyfikacja:
*Dane:
k* – liczba całkowita, *n* – liczba cyfr w zapisie dziesiętnym liczby *k*,cyfry[] – tablica zawierająca kolejne cyfry zapisu dziesiętnego liczby *k*, w kolejności od najmniej znaczącej do najbardziej znaczącej,potęga (i,n) – funkcja zwracająca *n*-tą potęgę liczby *i*.
*Wynik:* PRAWDA – jeśli dana liczba jest liczbą narcystyczną, lub FAŁSZ w przeciwnym wypadku.

Zadanie 2. Algorytm Babilon (0–6)

Poniżej przedstawiono pewien algorytm liczbowy, który w pierwotnej postaci został zapisany
na glinianej tabliczce znalezionej w Babilonie.

Krok 1: wczytaj *a*, *b*, *ε*
Krok 2: Δ:=|*a*–*b*2|
Krok 3: jeżeli Δ≤ *ε* skocz do Krok 6
Krok 4: *b*:=0,5(*b*+*a*/*b*)
Krok 5: skocz do Krok 2
Krok 6: wypisz *b*

a) Przeanalizuj powyższy algorytm i uzupełnij specyfikację i wynik. (0-1)

Specyfikacja:
*Dane:
a* - nieujemna liczba rzeczywista,
*b* - wstępna wartość oszacowania,
*ε* - ----

*Wynik:* nieujemna liczba rzeczywista, równa----

b) Niech *a* =8, *b* =1, *ε* =10 -6.
W poszczególnych wierszach w miejsce luk w kolumnie 2 wstaw odpowiednie wartości zmiennej *b*  i Δ w kolumnie 3 w czasie pierwszych 4 iteracji tego algorytmu. (0-1)
Zapisz numer luki i wartość poszczególnej zmiennej.

Legenda:

Wp –wartość początkowa
it –iteracja

Numer luki i wartość:

Np.: 1.1. 5, 1.2. 3 itd

Wp b =1 Δ = ----0.2.

it 1 b = ----1.1. Δ = ----1.2.

it 2 b = ----2.1. Δ = ----2.2.

it 3 b = ----3.1. Δ = ----3.2.

it 4 b = ----4.1. Δ = ----4.2.

c) Niech a =5, b =1.
 1. Dla podanego ε dopisz liczbę iteracji, po której zakończy się wykonywanie algorytmu.

Numer luki i wartość:

Np. 1 2, 2 5

 ε liczba iteracji

 1 ----1.
 0,5 ----2.
 0,25 ----3.
 0,125 ----4.
 0,0625 ----5.

2. Na podstawie punktu 1 wykonaj wykres zależności liczby iteracji od podanej wartości ε

3. Określ czy zdanie zapisane jako wniosek jest prawdziwe, czy fałszywe?
Zapisz obok wniosku P - jeśli zdanie jest prawdziwe lub F- jeśli zdanie jest fałszywe. (0-2).
Wniosek:Liczba iteracji oraz ε są do siebie wprost proporcjonalne

d) Dla bardzo małych wartości ε obliczenia mogą być długotrwałe, zmodyfikuj algorytm Babilon podany w tym zadaniu w taki sposób, aby jego wykonywanie zakończyło się po osiągnięciu zadanego ε lub po wykonaniu określonej liczby iteracji zapisanej w zmiennej *liczba\_iter.* (0–2)

Zadanie 3. (0–4)

Podpunkty z zadania 3 zawierają po trzy lub cztery odpowiedzi, z których każda jest albo prawdziwa, albo fałszywa.
Zapisz podpunkt i numer każdej odpowiedzi, a następnie P - jeśli zdanie jest prawdziwe lub F- jeśli zdanie jest fałszywe.

a) Niech a=(1001001)2, b=(211)­9, c=(211)­8 ­ wówczas:
1. b>c
2. a +b -c =0
3. c =(89)16

b) W sieciach komputerowych:
1. 192.168.0.1 jest adresem pętli zwrotnej
2. W klasie adresowej A mamy 27 adresów sieci i 224 adresów hostów.
3. Adresy 94.254.99.1/16 oraz 94.254.168.168/16 należą do jednej podsieci.

c) Protokołami służącymi do pobierania wiadomości elektronicznych ze zdalnego serwera są:
1. IMAP
2. SMTP
3. POP3
4. SNMP

d) W przypadku oprogramowania na licencji GNU GPL:
1. Możliwe jest wprowadzanie własnych poprawek
2. Oprogramowanie wyświetla w czasie pracy reklamy
3. Jest stosowana wyłącznie przy tworzeniu programów prototypowych, mogących działać niestabilnie.
4. Nie może być ono używane do działalności zarobkowej.