|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **WYPEŁNIA ZESPÓŁ NADZORUJĄCY** | | | | | | | | | | | | | | | | | ***Miejsce na naklejkę.***  *Sprawdź, czy kod na naklejce to* **E-660**. |
|  | | | | | | | | | | | | | | | | | |
| **KOD PESEL** | | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |

|  |
| --- |
| **EGZAMIN MATURALNY Z INFORMATYKI**  **Poziom rozszerzony**  **Część I**  **Test diagnostyczny**  Termin: **marzec 2021 r.**  Czas pracy: **do 90 minut**  Liczba punktów do uzyskania: **15** |

|  |  |
| --- | --- |
| **WYPEŁNIA ZDAJĄCY** | WYBRANE:  .................................................  (system operacyjny)  .................................................  (program użytkowy)  .................................................  (środowisko programistyczne) |

|  |
| --- |
| **Instrukcja dla zdającego**   1. Arkusz zawiera 3 zadania. 2. Obok każdego numeru zadania podana jest maksymalna liczba punktów, 3. którą można uzyskać za jego poprawne rozwiązanie . 4. Odpowiedzi zapisuj na kartkach dołączonych do arkusza, na których zespół nadzorujący wpisał Twój numer PESEL . 5. W razie pomyłki błędny zapis zapunktuj. 6. Napisz wybrane przez siebie środowisko programistyczne, system operacyjny i program użytkowy na kartę odpowiedzi. |

|  |
| --- |
| EINP-R1-**660**-2103 |
|
|
|

Zadanie 1. Turniej

W turnieju siatkówki bierze udział n drużyn ponumerowanych kolejnymi liczbami całkowitymi od 0 do n – 1, gdzie n = 2k dla pewnej liczby całkowitej k > 0. Turniej odbywa się w rundach systemem pucharowym – przegrywający odpada z turnieju. W każdej rundzie drużyny grają w parach i do dalszej rundy przechodzi tylko zwycięzca meczu. W każdej rundzie mecze są ponumerowane kolejnymi liczbami całkowitymi, poczynając od 1. W pierwszej rundzie w meczu nr 1 grają drużyny 0 i 1, w meczu nr 2 – drużyny 2 i 3, w meczu nr 3 – drużyny 4 i 5,   
w meczu nr i – drużyny 2\*(i – 1) oraz 2\*(i – 1)+1, itd. W każdej z kolejnych rund w meczu nr 1 grają zwycięzcy meczów o numerach 1 i 2 z poprzedniej rundy, w meczu nr 2 – zwycięzcy meczów o numerach 3 i 4 z poprzedniej rundy, w meczu nr i – zwycięzcy meczów o numerach 2\*i – 1 oraz 2\*i z poprzedniej rundy itd. Turniej trwa dokładnie k rund.

Przykład:

Przykładową rozgrywkę w turnieju 8-drużynowym przedstawiono w postaci drzewa na rysunku poniżej. Na najniższym poziomie rysunku drzewa zapisano numery drużyn, natomiast w węzłach wewnętrznych –numery zwycięskich drużyn w poszczególnych meczach. Zwycięzcą turnieju została drużyna nr 6, która w meczu finałowym pokonała drużynę o numerze 2.

runda 3 6

runda 2 2 6

runda 1 1 2 5 6

0 1 2 3 4 5 6 7

Numer rundy, w której mogą zmierzyć się dwie drużyny o numerach *x* i *y*, można wyznaczyć z zapisów binarnych liczb x i *y* o długości *k* (liczba rund). Twoim zadaniem jest odkrycie tej zależności.

Zadanie 1.1. (0–2)

Uzupełnij wiersze trzeci i czwarty - dla podanej liczby k (liczba rund w turnieju) oraz numerów drużyn x i y wyznacz numer rundy w turnieju, w której te dwie drużyny mogą się zmierzyć ze sobą.

Wiersz 1.

k =3, x =2, y =6, x dwójkowo =010, y dwójkowo =110, nr rundy, w której mogą się zmierzyć drużyny x i y: 3

Wiersz 2.

k =4, x =0, y =3, x dwójkowo =0000, y dwójkowo =0011, nr rundy, w której mogą się zmierzyć drużyny x i y: 2

Wiersz 3.

k =4, x =3, y =7, x dwójkowo =0011, y dwójkowo =0111, nr rundy, w której mogą się zmierzyć drużyny x i y: …

Wiersz 4.

k =5, x =16, y =30, x dwójkowo =10000, y dwójkowo =11110, nr rundy, w której mogą się zmierzyć drużyny x i y: …

Zadanie 1.2. (0–4)

Napisz algorytm (w pseudokodzie lub w wybranym języku programowania), który dla danych liczb całkowitych k, x i y obliczy numer rundy w turnieju dla 2k drużyn, w której mogą się spotkać drużyny x i y.

Uwaga: W zapisie algorytmu możesz korzystać wyłącznie z instrukcji sterujących, operatorów arytmetycznych (w tym dzielenia całkowitego i dzielenia z resztą), operatorów logicznych, porównań i instrukcji przypisywania lub samodzielnie napisanych funkcji i procedur. Zabronione jest używanie funkcji wbudowanych, dostępnych w językach programowania, a zwłaszcza funkcji podnoszącej do potęgi.

Specyfikacja algorytmu

Dane

k – dodatnia liczba całkowita, liczba rund w turnieju

x, y – dwie różne liczby całkowite z przedziału [0, 2k –1], numery drużyn

Wynik

runda – nr rundy, w której mogą się spotkać drużyny x i y

Algorytm: …

Zadanie 2. Analiza algorytmu

Wykonaj analizę funkcji Algo(n), której argumentem jest dodatnia liczba całkowita n.

Algo(n)

jeżeli n ≤2 to

wynikiem jest 1

w przeciwnym przypadku

p ← 1

k ← n

dopóki k – p >1 wykonuj

s ← (p + k) div 2

jeżeli s \* s ≤n to

p ← s

w przeciwnym przypadku

k ← s

wynikiem jest p

Uwaga: div oznacza dzielenie całkowite.

Zadanie 2.1. (0–2)

Uzupełnij wiersze drugi i trzeci – podaj wynik funkcji Algo dla podanych n.

Wiersz 1.

n =5, Algo(n) =2

Wiersz 2.

n =35, Algo(n) =…

Wiersz 3.

n =1025, Algo(n) =…

Zadanie 2.2. (0–3)

Uzupełnij wiersze drugi, trzeci i czwarty – podaj liczbę wykonań instrukcji „s ← (p + k) div 2” podczas obliczania wartości funkcji Algo(n) dla podanych wartości n.

Wiersz 1.

n =5, Odp. 2

Wiersz 2.

n =2, Odp. …

Wiersz 3.

n =63, Odp. …

Wiersz 4.

n =1024, Odp. …

Zadanie 3. Test

Oceń prawdziwość podanych zdań. Po numerze zdania zapisz P, jeśli zdanie jest prawdziwe, albo F – jeśli jest fałszywe. W każdym zadaniu punkt uzyskasz tylko za komplet poprawnych odpowiedzi.

Zadanie 3.1. (0–1)

W komórce C1 arkusza kalkulacyjnego zapisano formułę:

=JEŻELI(ORAZ(MOD(A1;2)=1;MOD(B1;2)=1);A1+B1;A1\*B1)

1. Jeśli w A1 wpisano liczbę 1, a w B1 liczbę 3, to w C1 w wyniku obliczenia formuły pojawi się liczba 4.

2. Jeśli w A1 wpisano liczbę 4, a w B1 liczbę 3, to w C1 w wyniku obliczenia formuły pojawi się liczba 3.

3. Jeśli w A1 i B1 wpiszemy dowolną liczbę całkowitą dodatnią, to w wyniku obliczenia formuły w C1 zawsze pojawi się liczba parzysta.

4. Jeśli w A1 i B1 wpiszemy dowolną liczbę całkowitą dodatnią, to w wyniku obliczenia formuły w C1 zawsze pojawi się liczba większa niż 1.

Zadanie 3.2. (0–1)

Mamy dane operacje (bramki) logiczne na bitach: notoraz and opisane poniżej:

|  |  |
| --- | --- |
| a | not a |
| 1 | 0 |
| 0 | 1 |

|  |  |  |
| --- | --- | --- |
| a | b | a and b |
| 1 | 1 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 0 | 0 | 0 |

oraz wyrażenie W(a,b): (not ((not a) and b)) and (not (a and (not b)))

Oceń prawdziwość podanych zdań. Po numerze zdania zapisz P, jeśli zdanie jest prawdziwe lub F – jeśli jest fałszywe.

1. W(0,0)=0

2. W(1,0)=0

3. W(0,1)=1

4. W(1,1)=1

Zadanie 3.3. (0–1)

Oceń prawdziwość podanych zdań. Po numerze zdania napisz P, jeśli zdanie jest prawdziwe lub F – jeśli jest fałszywe.

Różnica 10111012 – 101112 dwóch liczb zapisanych w systemie binarnym jest:

1. mniejsza niż 1001112

2. równa 10001102

3. większa niż 101112

4. równa 10010002

Zadanie 3.4. (0–1)

W bazie danych istnieje tabela oceny(id\_o, id\_u, przedmiot, ocena), zawierająca następujące dane:

|  |  |  |  |
| --- | --- | --- | --- |
| id\_o | id\_u | przedmiot | ocena |
| 1 | 1 | matematyka | 3 |
| 2 | 1 | informatyka | 4 |
| 3 | 1 | fizyka | 2 |
| 4 | 2 | matematyka | 6 |
| 5 | 2 | fizyka | 3 |
| 6 | 2 | informatyka | 5 |
| 7 | 3 | matematyka | 4 |
| 8 | 3 | fizyka | 2 |
| 9 | 3 | informatyka | 3 |

id\_o – identyfikator oceny, id\_u – identyfikator ucznia.

Oceń prawdziwość podanych zdań. Po numerze zdania napisz P, jeśli zdanie jest prawdziwe lub F – jeśli jest fałszywe.

1. Wynikiem zapytania: SELECT COUNT(id\_u) FROM oceny;

jest 3

2. Wynikiem zapytania: SELECT COUNT (id\_u) FROM oceny WHERE przedmiot="fizyka";

jest 3

3. Wynikiem zapytania: SELECT COUNT(przedmiot) FROM oceny;

jest 9

4. Wynikiem zapytania: SELECT COUNT(przedmiot) FROM oceny WHERE ocena >3;

jest 4