**Arkusz I 2020 - Klucz rozwiązań**

**Zadanie 1.1 (0-1)**

**Poprawna odpowiedź: F, F, P, F**

**Zadanie 1.2 (0-1)**

**Poprawna odpowiedź:**

|  |  |
| --- | --- |
| Polecenie | Definicja |
| 1. | **A** |
| 2. | **B** |
| 3. | **D** |
| 4. | **C** |

**Zadanie 1.3. (0–1)**

**Poprawna odpowiedź: P, P, F, P**

**Zadanie 1.4. (0–1)**

**Poprawna odpowiedź: F, P, F, F**

**Zadanie 1.5. (0–1)**

**Poprawna odpowiedź: P, P, F, P.**

**Zadanie 2. (0–5)**

**Zadanie 2.1 (0-2)**

|  |  |  |
| --- | --- | --- |
| **Dane** | **Para liczb względnie pierwszych**  **TAK / NIE** | **Liczba operacji** |
| , | TAK | 4 |
| , | NIE | 2 |
| , | TAK | 3 |
| , , gdzie oznacza -tą liczbę Fibonacciego | TAK |  |

**1 p.** za poprawne wypełnienie wierszy 2 i 3,

**1 p.** za poprawne wypełnienie wiersza 4.

**Zadanie 2.2. (0–3)**

**Lista kroków:**

**Krok 1.**

**Krok 2.** Dla wykonuj Krok 2.1

**Krok 2.1.** Dla wykonuj Krok 2.1.1

**Krok 2.1.1.** Jeżeli , to

**Krok 3.** Wypisz i zakończ algorytm

**3 p.** przyznajemy za w całości poprawne rozwiązanie

**2 p.** za rozwiązanie z niepotrzebnym zliczaniem par

**1 p.** za rozwiązanie ze zliczaniem wszystkich możliwych par (bez rozróżnienia par

**Zadanie 3 (0-5)**

**Schemat oceniania**

|  |  |  |  |
| --- | --- | --- | --- |
| Nr pytania | Oczekiwana odpowiedź | Maksymalna punktacja | |
| 3.1 | Za podanie pełnej poprawnej odpowiedzi – 2 punkty,  przy czym po 1 punkcie za poprawne odpowiedzi dla każdego ciągu  Odpowiedzi:  Pierwszy ciąg:  5 – 3 = 2  -3 + 2 - 5 = -6  Drugi ciąg:  6 + 1 + 2 + 9 = 18  9 + 3 + 4 = 16  2 + 6 + 1 + 2 = 11 | 2 | 5 |
| 3.2 | Za podanie poprawnego algorytmu o złożoności co najwyżej O(n + m) – 3 pkt.  Za podanie poprawnego algorytmu o gorszej złożoności (np. O(n\*m)) – 2 pkt. | 3 |

**Przykładowe rozwiązanie o złożoności O(n + m)**

#include <iostream>

using namespace std;

int main(){

int n,a,b; cin>>n;

int k[n+1];

int p[n+1];

p[0]=0;

for (int i=1;i<=n; ++i){

cin>>k[i];

p[i]=p[i-1]+k[i];

}

int m; cin>>m;

for (int i=0; i<m; ++i){

cin>>a; cin>>b;

cout<<p[b]-p[a-1]<<endl;

}

return 0;

}

**Przykładowe rozwiązanie o złożoności O(n \* m)**

int main()

{ int n,a,b; cin>>n;

int k[n+1];

int p[n+1];

p[0]=0;

for (int i=1;i<=n; ++i) cin>>k[i];

int m; cin>>m;

for (int i=0; i<m; ++i)

{ int s=0;

cin>>a; cin>>b;

for (int j=a; j<=b; ++j) s=s+k[i];

cout<<s;

}

return 0;

}