
Arkusz I 2021 - Klucz rozwiązań

Uwaga: Odwołania do Aneksu 2021 są poprzedzone literą A.

Zadanie 1.1 (0-1)

Poprawna odpowiedź: F,F,P,F

Wymagania ogólne	Wymagania szczegółowe
III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	Zdający analizuje i rozwiązuje sytuacje problemowe z różnych dziedzin (5.1, A4.1), Zdający stosuje podejście algorytmiczne do rozwiązywania problemu (5.2, A4.2), stosuje rekurencję w prostych sytuacjach problemowych (5.9, A4.8), opisuje własności algorytmów na podstawie ich analizy (16, A15)

Zadanie 1.2 (0-1)

Poprawna odpowiedź: F, P, F, F

Wymagania ogólne	Wymagania szczegółowe
III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	Zdający analizuje i rozwiązuje sytuacje problemowe z różnych dziedzin (5.1, A 4.1), Zdający stosuje podejście algorytmiczne do rozwiązywania problemu (5.2, A 4.1), opisuje podstawowe algorytmy i stosuje algorytmy wyszukiwania i porządkowania (5.11.b, A 4.10.b)

Zadanie 1.3. (0–1)

Poprawna odpowiedź: P, P, F, F

Wymagania ogólne	Wymagania szczegółowe
III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	Zdający analizuje i rozwiązuje sytuacje problemowe z różnych dziedzin (5.1, A 4.1), Zdający stosuje podejście algorytmiczne do rozwiązywania problemu (5.2, A 4.2), dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne (14, A 13),) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania (5.5, A 4.5);

Zadanie 1.4. (0–1)**Poprawna odpowiedź: P, F, F, P**

Wymagania ogólne	Wymagania szczegółowe
I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.	Zdający korzysta z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją; (1.2, A 1.2).

Zadanie 1.5. (0–1)**Poprawna odpowiedź: F, P, F, F.**

Wymagania ogólne	Wymagania szczegółowe
I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.	Zdający przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków (1.1, A 1.1).

Zadanie 2. (0–5)

Wymagania ogólne	Wymagania szczegółowe
III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego	Zdający stosuje podejście algorytmiczne do rozwiązywania problemu (5.2, A 4.2), opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania (5.7, A 4.6), stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu (5.13, A.4.12)

Zadanie 2.1. (1 punkt)

1 p. za podanie wszystkich poprawnych wartości punktów i zwycięzcy.

0 p. za odpowiedź niepełną lub błędną albo za brak odpowiedzi.

Poprawna odpowiedź:

Strategia Jasia	Strategia Małgosi	Punkty Jasia	Punkty Małgosi	Zwycięzca
A	A	28 (10+8+6+4)	21 (9+7+5)	Małgosia
B	B	25 (1+3+5+7+9)	30 (2+4+6+8+10)	Jaś
A	B	27 (10+9+8)	3 (1+2)	Małgosia
B	A	6 (1+2+3)	27 (10+9+8)	Jaś

Zadanie 2.2. (2 punkty)

1 p. za poprawne funkcje zwracające wartość kolejnego żetonu do wybrania

0 p. za funkcje błędne, niedokończone lub brak rozwiązania

Przykładowe rozwiązanie:**Lista kroków:**

stratA(zetony, n):

1. maks := 0
2. Dla każdego żetonu zet w tablicy zetony, wykonaj:
 - 2.1. Jeżeli zet > maks, to maks := zet
3. Zwróć maks

stratB(zetony, n):

1. min := 0
2. Dla każdego żetonu zet w tablicy zetony, wykonaj:
 - 2.1. Jeżeli zet < min, to min := zet
3. Zwróć min

Python:

```
def stratA(zetony, n):
    maks = 0
    for zet in zetony:
        if zet > maks:
            maks = zet
    return maks

def stratB(zetony, n):
    min = 0
    for zet in zetony:
        if zet < min:
            min = zet
    return min
```

C++

```
int stratA(int zetony[], int n) {
    int maks = 0;
    for(int i = 0; i < n; i++) {
        if(zetony[i] > maks) {
            maks = zetony[i];
        }
    }
    return maks;
}

int stratB(int zetony[], int n) {
    int min = 0;
    for(int i = 0; i < n; i++) {
        if(zetony[i] < min) {
            min = zetony[i];
        }
    }
    return min;
}
```

C++ 11

```
int stratA(int zetony[], int n) {
    int maks = 0;
    for(int zet : zetony) {
        if(zet > maks) {
            maks = zet;
        }
    }
    return maks;
}

int stratB(int zetony[], int n) {
    int min = 0;
    for(int zet : zetony) {
        if(zet < min) {
            min = zet;
        }
    }
    return min;
}
```

Zadanie 2.3. (3 punkty)

3 p. za poprawny program znajdujący zwycięzcę dla podanych strategii, w tym:

1 p. za poprawny sposób obliczania sumy zdobytych punktów

1 p. za poprawne wykorzystanie funkcji stratA oraz stratB

1 p. za poprawne sprawdzenie, czy dany gracz przekroczył 25 punktów i wypisanie zwycięzcy

0 p. za rozwiązanie błędne lub brak rozwiązania

Przykładowe rozwiązanie:

Lista kroków:

1. zetony := [1,2,3,4,5,6,7,8,9,10]
2. n := 10
3. sumaJ := 0
4. sumaM := 0
5. ruchJ := 0, ruchM := 0
6. Powtarzaj:
 - 6.1. Jeżeli strategiaJasia = 'A', wykonaj ruchJ := stratA(zetony, n)
 - 6.2. W przeciwnym przypadku, wykonaj ruchJ := stratB(zetony, n)
 - 6.3. sumaJ := sumaJ + ruchJ
 - 6.4. usun(zetony, ruchJ)
 - 6.5. n := n - 1
 - 6.6. Jeżeli sumaJ > 25, to wypisz „Malgosia” i zakończ
 - 6.7. Jeżeli strategiaMalgosi = 'A', wykonaj ruchM := stratA(zetony, n)
 - 6.8. W przeciwnym przypadku, wykonaj ruchM := stratB(zetony, n)
 - 6.9. sumaM := sumaM + ruchM
 - 6.10. usun(zetony, ruchM)
 - 6.11. n := n - 1
 - 6.12. Jeżeli sumaM > 25, to wypisz „Jas” i zakończ

Python:

```
zetony = [1,2,3,4,5,6,7,8,9,10]
n = 10
sumaJ = 0
sumaM = 0
ruchJ = 0
ruchM = 0
while True:
    if strategiaJasia == „A”:
        ruchJ = stratA(zetony, n)
    else:
        ruchJ = stratB(zetony, n)
    sumaJ += ruchJ
    usun(zetony, ruchJ)
    n -= 1
    if sumaJ > 25:
        print(„Malgosia”)
        break
    if strategiaMalgosi == „A”:
        ruchM = stratA(zetony, n)
    else:
        ruchM = stratB(zetony, n)
    sumaM += ruchM
    usun(zetony, ruchM)
    n -= 1
    if sumaM > 25:
```

```
print(„Jas”)
break
```

C++

```
#include<iostream>
using namespace std;

int main() {
int zetony[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int n = 10;
int sumaJ = 0, sumaM = 0, ruchJ = 0, ruchM = 0;
while(true){
    if(strategiaJasia == 'A'){
        ruchJ = stratA(zetony, n);
    } else {
        ruchJ = stratB(zetony, n);
    }
    sumaJ += ruchJ;
    usun(zetony, ruchJ);
    n--;
    if(sumaJ > 25){
        cout << „Malgosia” << endl;
        break;
    }
    if(strategiaMalgosi == 'A'){
        ruchM = stratA(zetony, n);
    } else {
        ruchM = stratB(zetony, n);
    }
    sumaM += ruchM;
    usun(zetony, ruchM);
    n--;
    if(sumaM > 25){
        cout << „Jas” << endl;
        break;
    }
}
} }
```

Zadanie 3 (0-5)

Wymagania ogólne	Wymagania szczegółowe
III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.	Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. (5, A 4) Zdający analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin (5.1, A 4.1), stosuje algorytmiczne podejście do rozwiązywania problemu (5.2, A 4.2),

	dobiera efektywne algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji (5.4, A 4.4)
--	--

Zadanie 3.1 [1 pkt]

1 punkt za podanie poprawnej odpowiedzi

0 punktów za odpowiedź niepełną, błędną lub za brak odpowiedzi

Poprawna odpowiedź:

x – ilość liczb parzystych w pierwszej kolumnie tablicy A

Zadanie 3.2 (2 punkty)

2 p. za poprawny algorytm o złożoności $O(N)$

1 p. za poprawny algorytm o złożoności większej niż $O(N)$

0 p. za nie działający poprawnie algorytm, odpowiedź niepełną lub jej brak

Przykładowe rozwiązanie:

Lista kroków:

Krok 1. Rozpocznij algorytm

Krok 2. $x \leftarrow 0$

Krok 3. Dla $i \leftarrow 1, 2, \dots, N$ wykonuj Krok 4:

Krok 4. $x \leftarrow x + [1 - (a_{i,1} \bmod 2)]$

Krok 5. Wypisz x jako wynik i zakończ algorytm.

Zadanie 3.3 (2 punkty)

2 punkty za poprawny program o podanej złożoności, w tym:

1 punkt za złożoność $O(N \log M)$

1 punkt za poprawne wyszukiwanie wartości x

0 punktów za rozwiązanie błędne lub jego brak

Przykładowe rozwiązanie:

Lista kroków:

Krok 1. Rozpocznij algorytm

Krok 2. $L \leftarrow 0$

Krok 3. Dla $i \leftarrow 1, 2, \dots, N$ wykonuj Kroki 4-9:

Krok 4. $\text{czy_wyst} \leftarrow 0, le \leftarrow 1, pr \leftarrow M$

Krok 5. Dopóki $le < pr$ wykonuj Kroki: 6-8

Krok 6. $s \leftarrow (le + pr) \text{ div } 2$

Krok 7. Jeżeli $A_{i,s} = x$, to $\text{czy_wyst} \leftarrow 1$ i przejdź do Kroku 9

Krok 8. Jeżeli $x < A_{i,s}$, to $pr \leftarrow s-1$, a w przeciwnym przypadku $le \leftarrow s+1$

Krok 9. $L \leftarrow L + \text{czy_wyst}$

Krok 10. Wypisz L jako wynik i zakończ algorytm.