



Rekurencja – zadania maturalne

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Źródło: Pixabay, domena publiczna.

W e-materiale [Rekurencja](#) poznaliśmy zagadnienie kluczowe w rozwiązywaniu wielu problemów informatycznych. Rekurencji, czyli metody rozwiązywania problemów, polegającej na rozwiązywaniu podproblemów, możemy użyć m.in. w programach generujących ciągi liczb, wyszukujących wartości, sortujących zbiory czy generujących fraktale. W tym e-materiale rozwiązaliśmy zadania naturalne, w których należy wykorzystać rekurencję.

O tym, jak zagadnienie rekurencji wyjaśnia matematyka, przeczytasz w e-materiałach:

- [Ciąg określony rekurencyjnie](#),
- [Ciąg geometryczny określony rekurencyjnie](#),
- [Wzór ogólny ciągu określonego rekurencyjnie](#),
- [Ciąg arytmetyczny określony wzorem rekurencyjnym](#).

Implementację rekurencji przedstawiamy w e-materiałach:

- [Rekurencja w języku C++](#),
- [Rekurencja w języku Java](#),
- [Rekurencja w języku Python](#).

Więcej zadań? Sięgnij do: [Rekurencja – ćwiczenia](#).

Twoje cele

- Prześledzisz sposób rozwiązywania zadań maturalnych wykorzystujących rekurencję.
- Rozwiążesz samodzielnie kilka zadań maturalnych, używając rekurencji.
- Przeanalizujesz, w jaki sposób uniknąć typowych błędów w tego typu zadaniach.

Przeczytaj

Rekurencja to technika rozwiązywania problemów, będąca alternatywą dla metody iteracyjnej. W podejściu rekurencyjnym funkcja rozwiązująca problem wywołuje samą siebie, aż do napotkania tzw. [przypadku podstawowego](#).

Przykładowe zadania maturalne

Zadanie 1

Oceń, czy przedstawione zdania są prawdziwe. Zaznacz P, jeśli zdanie jest prawdziwe, albo F – jeśli zdanie jest fałszywe.

Zadanie 1.1

Dana jest funkcja f określona wzorem rekurencyjnym:

$$f(1) = 4$$

$$f(n+1) = \frac{1}{1-f(n)} \quad \text{dla } n \geq 1$$

Wtedy:

L.p	Funkcja	Czy prawda	Czy fałsz
1.	$f(8) = \frac{1}{3}$	P	F
2.	$f(9) = \frac{3}{4}$	P	F
3.	$f(10) = 4$	P	F
4.	$f(100) = -\frac{1}{3}$	P	F

Zadanie zostało opracowane przez Centralną Komisję Egzaminacyjną i pojawiło się na egzaminie maturalnym z informatyki w 2016 roku (poziom rozszerzony, egzamin w tzw. starej formule). Cały arkusz można znaleźć na stronie internetowej CKE.

Przykładowe rozwiązanie

To zadanie sprawdza umiejętności czytania i rozumienia algorytmów rekurencyjnych. Przypadkiem podstawowym jest tu wartość $x = 1$, dla której funkcja f zwraca wartość 4.

Przeanalizujemy działanie tego algorytmu dla każdego z czterech przypadków.

Przypadek 1: $f(8)$

$$f(8) = \frac{1}{1 - f(7)}$$

$$f(7) = \frac{1}{1 - f(6)}$$

$$f(6) = \frac{1}{1 - f(5)}$$

$$f(5) = \frac{1}{1 - f(4)}$$

$$f(4) = \frac{1}{1 - f(3)}$$

$$f(3) = \frac{1}{1 - f(2)}$$

$$f(2) = \frac{1}{1 - f(1)}$$

$$f(1) = 4$$

$$f(2) = \frac{1}{1 - 4} = \frac{1}{-3} = -\frac{1}{3}$$

$$f(3) = \frac{1}{1 - \left(-\frac{1}{3}\right)} = \frac{1}{1\frac{1}{3}} = \frac{3}{4}$$

$$f(4) = \frac{1}{1 - \frac{3}{4}} = \frac{1}{\frac{1}{4}} = 4$$

$$f(5) = \frac{1}{1 - 4} = \frac{1}{-3} = -\frac{1}{3}$$

$$f(6) = \frac{1}{1 - (-\frac{1}{3})} = \frac{1}{1\frac{1}{3}} = \frac{3}{4}$$

$$f(7) = \frac{1}{1 - \frac{3}{4}} = \frac{1}{\frac{1}{4}} = 4$$

$$f(8) = \frac{1}{1 - 4} = \frac{1}{-3} = -\frac{1}{3}$$

Otrzymany wynik jest inny niż podany, więc odpowiedzią jest F.

Przypadek 2: $f(9)$

$$f(9) = \frac{1}{1 - (-\frac{1}{3})} = \frac{1}{1\frac{1}{3}} = \frac{3}{4}$$

Otrzymany wynik jest taki sam jak podany, więc odpowiedzią jest P.

Przypadek 3: $f(10)$

$$f(10) = \frac{1}{1 - \frac{3}{4}} = \frac{1}{\frac{1}{4}} = 4$$

Otrzymany wynik jest taki sam jak podany, więc odpowiedzią jest P.

Przypadek 4: $f(100)$

Tego przykładu nie będziemy rozpisywać. Jest on zbyt obszerny, a rozwiązanie można podać o wiele szybciej. Zauważmy bowiem, że w wynikach powtarza się pewien schemat występowania liczb. Pierwszym wynikiem (dla $f(1)$) jest 4, drugim $-\frac{1}{3}$, trzecim $\frac{3}{4}$,

a czwartym ponownie 4. Dla kolejnych wyników występuje ten sam schemat – każda wartość powtarza się co 3 wyniki.

Obliczmy zatem wartość $f(100)$. Pierwszym wynikiem jest liczba 4. Powtarza się ona co 3, więc taki wynik będzie kolejno dla $f(4)$, $f(7)$, $f(10)$, $\dots$, $f(100)$.

Dochodzimy w ten sposób do funkcji $f(100)$, której wartość jest równa 4.

Otrzymany wynik jest inny niż podany, więc odpowiedzią jest F.

Drugi sposób: wystarczy obliczyć resztę z dzielenia $x = x \bmod 3$. W omawianym przypadku $x = 100$, wynik dzielenia z resztą to $x = 1$ czyli $f(x) = 4$.

Schemat oceniania

I. Wiadomości i rozumienie	Zdający zna techniki algorytmiczne (rekurencja)
----------------------------	---

Schemat punktowania

1 pkt – za wszystkie cztery poprawne odpowiedzi

0 pkt – za odpowiedź niepełną lub błędną albo brak odpowiedzi

Poprawna odpowiedź

F, P, P, F

Schemat oceniania pochodzi z arkusza odpowiedzi wykorzystywanego podczas egzaminu maturalnego z informatyki w 2016 roku (na poziomie rozszerzonym). Cały arkusz można znaleźć na stronie internetowej Centralnej Komisji Egzaminacyjnej.

Praca domowa

Napisz implementację omówionej [funkcji rekurencyjnej](#) w wybranym języku programowania.

Zadanie 2. Długość napisów binarnych (7 pkt.)

Treść zadania

Opisana poniżej funkcja rekurencyjna wyznacza dla liczby naturalnej $n > 0$ długość napisu uzyskanego przez sklejenie binarnych reprezentacji liczb naturalnych od 1 do $n - 1$.

Funkcja $sklej(n)$

Krok 1. Jeśli $n = 1$, to podaj 0 jako wynik i zakończ działanie.

Krok 2. Jeśli n parzysta, to wynikiem jest $n - 1 + 2 \cdot \text{sklej}(n : 2)$.

Krok 3. Jeśli n jest nieparzysta, to wynikiem jest $n - 1 + \text{sklej}((n - 1) : 2) + \text{sklej}((n + 1) : 2)$.

b) Uzupełnij tabelę, podając wartości funkcji sklej dla wskazanych argumentów.

n	sklej(n)
1	0
2	1
3	
4	
5	
6	

Zadanie zostało opracowane przez Centralną Komisję Egzaminacyjną i pojawiło się na egzaminie maturalnym z informatyki w 2011 roku (poziom rozszerzony). Cały arkusz można znaleźć na stronie internetowej CKE.

Rozwiązanie podpunktu **a)** tego zadania zostało omówione w sekcji „Prezentacja multimedialna”.

Przykładowe rozwiązanie:

Obliczmy kolejne wartości podane w tabeli.

$$\text{sklej}(3) = 3 - 1 + \text{sklej}\left(\frac{(3 - 1)}{2}\right) + \text{sklej}\left(\frac{(3 + 1)}{2}\right) =$$

$$= 2 + \text{sklej}(1) + \text{sklej}(2) = 2 + 0 + 1 = 3$$

$$\text{sklej}(4) = 4 - 1 + 2 \cdot \text{sklej}\left(\frac{4}{2}\right) = 3 + 2 \cdot \text{sklej}(2) =$$

$$= 3 + 2 \cdot 1 = 3 + 2 = 5$$

$$sklej(5) = 5 - 1 + sklej\left(\frac{(5-1)}{2}\right) + sklej\left(\frac{(5+1)}{2}\right) =$$

$$4 + sklej(2) + sklej(3) = 4 + 1 + 3 = 8$$

$$sklej(6) = 6 - 1 + 2 \cdot sklej\left(\frac{6}{2}\right) = 5 + 2 \cdot sklej(3) =$$

$$= 5 + 2 \cdot 3 = 5 + 6 = 11$$

Schemat oceniania

Korzystanie z informacji	Obliczenie kolejnych wartości funkcji dla wskazanych argumentów
--------------------------	---

Poprawna odpowiedź

n	sklej(n)
1	0
2	1
3	3
4	5
5	8
6	11

2 pkt – za poprawne uzupełnienie wartości funkcji w tabeli

1 pkt – za uzupełnienie wartości funkcji w tabeli z jednym błędem

0 pkt – za wypełnioną tabelę z więcej niż jednym błędem albo brak odpowiedzi

Schemat oceniania pochodzi z arkusza odpowiedzi wykorzystywanego podczas egzaminu maturalnego z informatyki w 2011 roku (na poziomie rozszerzonym). Cały arkusz można znaleźć na stronie internetowej Centralnej Komisji Egzaminacyjnej.

Zwróćmy uwagę na schemat oceniania w tym zadaniu. Za prawidłowo wypełnioną tabelę uzyskamy 2 punkty. Gdy jedna z wartości będzie błędna, otrzymamy 1 punkt. Z tego powodu zawsze warto sprawdzić poprawność otrzymywanych wyników.

Słownik

funkcja rekurencyjna

funkcja odwołująca się do samej siebie w celu rozwiązania problemu aż do napotkania przypadku podstawowego

przypadek podstawowy

przypadek, w którym funkcja rekurencyjna zwraca konkretną wartość, a nie wywołanie samej siebie

Prezentacja multimedialna

Problem 1

Przeanalizuj działanie funkcji `sklej (n)`, a następnie rozwiąż zadanie.

Funkcja `sklej (n)`

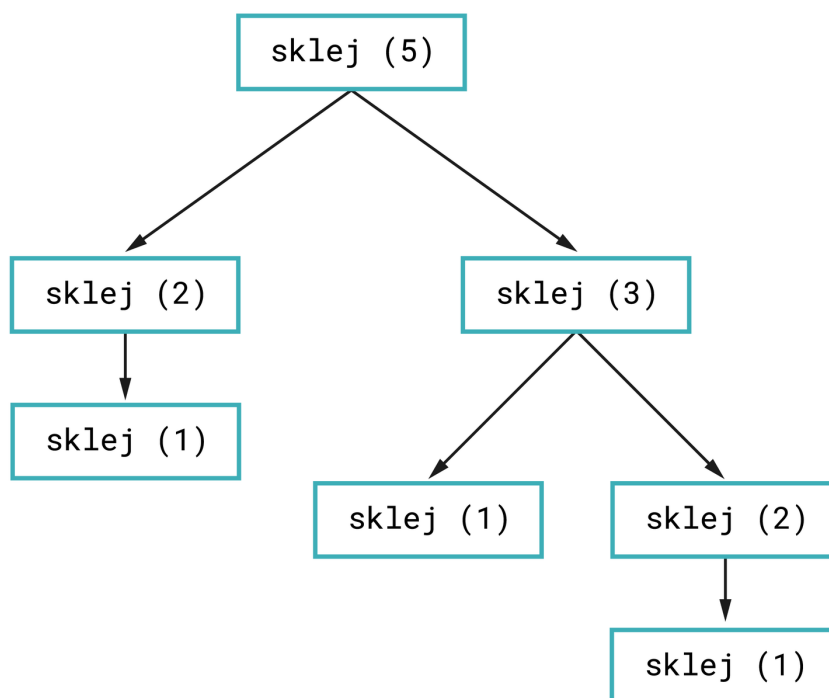
Krok 1. Jeśli $n = 1$, to podaj 0 jako wynik i zakończ działanie.

Krok 2. Jeśli n parzysta, to wynikiem jest $n - 1 + 2 \cdot \text{sklej}(n : 2)$.

Krok 3. Jeśli n jest nieparzysta, to wynikiem jest $n - 1 + \text{sklej}((n - 1) : 2) + \text{sklej}((n + 1) : 2)$.

a) Wykonanie funkcji `sklej` można przedstawić w postaci drzewa wywołań rekurencyjnych, ilustrującego wszystkie wywołania funkcji po jej uruchomieniu dla zadanego argumentu.

Rysunek przedstawia takie drzewo dla wywołania `sklej (5)`.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Polecenie 1

Narysuj podobne drzewo dla wywołania `sklej (7)`.

Zadanie zostało opracowane przez Centralną Komisję Egzaminacyjną i pojawiło się na egzaminie maturalnym z informatyki w 2011 roku (poziom rozszerzony, część I). Cały arkusz można znaleźć na stronie internetowej CKE.

Polecenie 2

Porównaj swoje rozwiązanie z prezentacją, w której krok po kroku przedstawiono tworzenie schematu reprezentującego drzewo wywołań rekurencyjnych dla funkcji `sklej (7)`.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PVZfv38xl>

Naszym zadaniem jest stworzyć schemat reprezentujący drzewo wywołań rekurencyjnych dla funkcji `sklej (7)`.

1

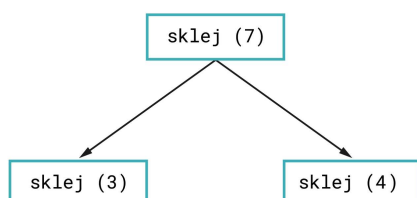
2

`sklej (7)`

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PVZfv38xl>

Zaczynamy od pierwszego bloku z wywołaniem `sklej (7)`.



3

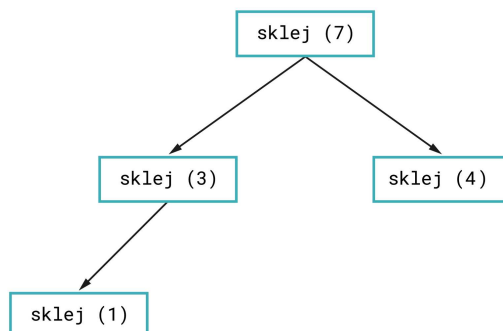
Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PVZfv38xl>

Liczba 7 jest nieparzysta, więc – zgodnie z algorytmem – w tym przypadku wywołane

zostaną kolejne dwie funkcje: `sklej (3)` oraz `sklej (4)`.

4

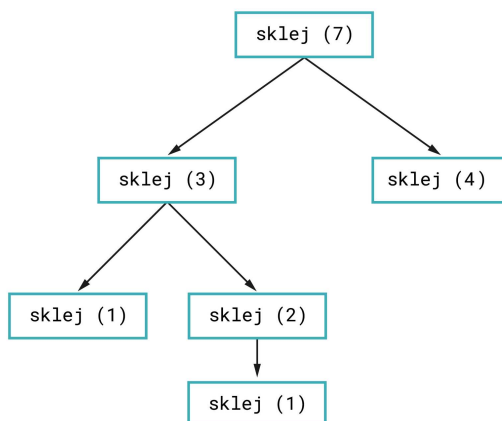


Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PVZfv38xl>

Przejdźmy najpierw do wywołania `sklej (3)`. Wartość 3 też jest nieparzysta, a więc znowu zostaną wywołane kolejne dwie funkcje. Jedną z nich jest `sklej (1)`.

5

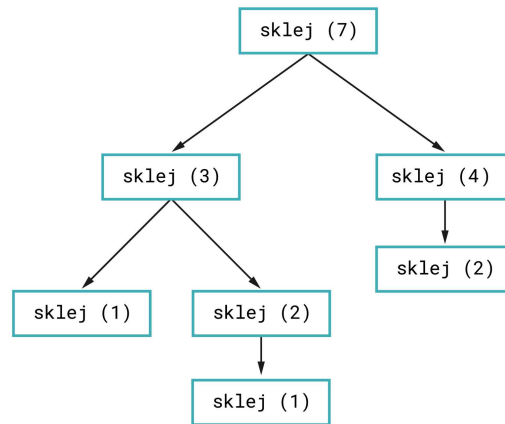


Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PVZfv38xl>

Drugą z kolei jest funkcja `sklej (2)`, która następnie wywołuje funkcję `sklej (1)`.

6

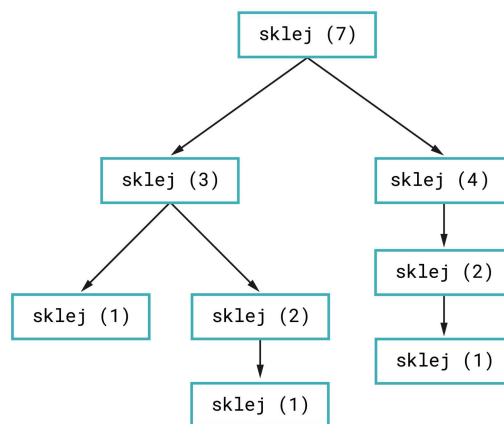


Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PVZfv38xl>

Powracamy do wywołanej w kroku 3 naszego rozwiązania funkcji `sklej (4)`, której wykonanie wywołuje jedną funkcję `sklej (2)`.

7



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PVZfv38xl>

Ostatnią funkcją wywołaną w tym drzewie będzie kolejna funkcja `sklej (1)`.

Za prawidłowe drzewo wywołań funkcji `sklej` dostajemy 1 pkt, w innym przypadku dostajemy 0 pkt. Schemat oceniania pochodzi z arkusza odpowiedzi wykorzystanego podczas egzaminu maturalnego z informatyki w 2011 roku (na poziomie rozszerzonym). Cały arkusz można znaleźć na stronie internetowej Centralnej Komisji Egzaminacyjnej.

Sprawdź się

Treść do ćwiczeń 1-3:

Dana jest funkcja `znajdź`, opisana za pomocą niepełnej specyfikacji i pseudokodu.

Specyfikacja:

Dane:

- n – liczba naturalna
- `tab[1..n]` – tablica n liczb rzeczywistych

Wynik:

...

Pseudokod:

```
1 funkcja znajdź(n, tablica):  
2     jeżeli n = 1  
3         zwróć tablica[1]  
4  
5     wynik ← znajdź(n - 1, tablica)  
6     jeżeli wynik > tablica[n]  
7         zwróć wynik  
8     zwróć tablica[n]
```

Na podstawie podanych informacji wykonaj następujące ćwiczenia:

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż wynik funkcji `znajdź(n, tab)`, gdzie `tab` to tablica n liczb rzeczywistych.

☐ najmniejsza spośród liczb `tab[1..n]`

☐ średnia liczb zawartych w `tab`

☐ największa spośród liczb `tab[1..n]`

Ćwiczenie 2



Wskaż wszystkie prawidłowe odpowiedzi. Operacją dominującą funkcji `znajdź` jest porównanie dwóch wartości występujących w linii 6. algorytmu. Zaznacz zdania dotyczące liczby operacji dominujących tej funkcji.

☐

dla `znajdź(10, [1, 2, 3, 4, 5, 6, 7, 8, 9, 0])` liczba dominujących operacji wynosi 9

☐

dla `znajdź(4, [143, 143, 143, 143])` liczba dominujących operacji wynosi 3

☐

dla `znajdź(10, [1, 2, 3, 4, 5, 6, 7, 8, 9, 0])` liczba dominujących operacji wynosi 10

☐

dla `znajdź(4, [143, 143, 143, 143])` liczba dominujących operacji wynosi 1

☐

dla `znajdź(10, [0, 9, 8, 7, 6, 5, 4, 3, 2, 1])` liczba dominujących operacji wynosi 1

☐

dla `znajdź(5, [99, 0, 0, 0, 0])` liczba dominujących operacji wynosi 2

☐

dla `znajdź(5, [99, 0, 0, 0, 0])` liczba dominujących operacji wynosi 4

Ćwiczenie 3



Wskaż prawidłowe dokończenie zdania.

Złożoność obliczeniowa funkcji `znajdź` jest...

☐

liniowa, opisana wzorem $O(n+k)$, gdzie n to rozmiar tablicy `tab`, a k to różnica pomiędzy największym i najmniejszym elementem w tablicy.

☐

kwadratowa, opisana wzorem $O(k^2)$, gdzie k oznacza pozycję mediany tablicy `tab`.

☐

liniowa, opisana wzorem $O(n)$, gdzie n to rozmiar tablicy `tab`.

☐

stała.

Ćwiczenie 4



Zapisz za pomocą pseudokodu funkcję rekurencyjną `wypisz(długość, tekst)`, która wypisze liczby binarne w przedziale $<0, 2^{\text{długość}-1}>$. Liczby dopełnij zerami z lewej strony w taki sposób, żeby wszystkie były takiej samej długości. Przyjmij, że wartość zmiennej `tekst` w początkowym wywołaniu to pusty łańcuch znaków. Przykładowe wywołanie funkcji `wypisz(2, " ")` powinno zwrócić następujący wynik:

```
1 00
2 01
3 10
4 11
```

Do dyspozycji masz funkcję `wydrukuj`, która wypisuje tekst na ekranie, długość, która zwraca długość łańcucha tekstowego, a także operację konkatencji tekstów, czyli ich łączenia, zapisaną za pomocą znaku plus.

```
1
```



Problem 1

Liczby Stirlinga II rodzaju mówią o liczbie sposobów, na jaki możemy podzielić zbiór rozróżnialnych niepowtarzalnych elementów (np. osoby z jednej klasy) na niepuste nierozróżnialne podzbiory (np. na grupy projektowe), przy czym liczebność tych podzbiorów jest ograniczona jedynie od dołu przez 1.

Jednym ze sposobów wyznaczania liczb Stirlinga II rodzaju jest poniższy wzór rekurencyjny (czytamy „ k podzbiorów n ”):

$$\left\{ \begin{matrix} n \\ k \end{matrix} \right\} = k \cdot \left\{ \begin{matrix} n-1 \\ k \end{matrix} \right\} + \left\{ \begin{matrix} n-1 \\ k-1 \end{matrix} \right\}$$

gdzie n to liczba elementów w zbiorze, a k to liczba podzbiorów, na które chcemy podzielić wyjściowy zbiór. Warunki początkowe prezentują się następująco:

$$\left\{ \begin{matrix} n \\ 1 \end{matrix} \right\} = 1, \quad \left\{ \begin{matrix} n \\ 0 \end{matrix} \right\} = 0, \quad \left\{ \begin{matrix} n \\ n \end{matrix} \right\} = 1$$

Ponadto, jeżeli $k > n$ to $\left\{ \begin{matrix} n \\ k \end{matrix} \right\} = 0$.

Napisz program, który oblicza rekurencyjnie wartość liczby Stirlinga II rodzaju dla podanych liczebności zbioru n i liczby podzbiorów k . Przetestuj swój program dla liczby Stirlinga $\left\{ \begin{matrix} 7 \\ 4 \end{matrix} \right\}$.

Zrealizuj zadanie w jednym z dostępnych języków programowania.

Specjalizacja:

Dane:

- n – liczba naturalna oznaczająca liczebność zbioru wyjściowego
- k – liczba naturalna określająca, na ile podzbiorów dokonujemy podziału

Wynik:

Program na standardowym wyjściu wypisze wartość liczby Stirlinga II rodzaju $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$.



Twoje zadania

1. Program w języku Java na standardowym wyjściu wypisuje wartość liczby Stirlinga II rodzaju.

```
1 public class Stirling {
2     static int stirling(int n, int k) {
3         return 0;
4     }
5
6     public static void main(String[] args) {
7         int n = 7;
8         int k = 4;
9
10        System.out.println(stirling(n, k));
11    }
12 }
```

```
1
```





JĘZYK C++

Twoje zadania

1. Program w języku C++ na standardowym wyjściu wypisuje wartość liczby Stirlinga II rodzaju.

```
1 #include <iostream>
2 using namespace std;
3
4 int stirling(int n, int k) {
5     return 0;
6 }
7
8 int main() {
9     int n = 7;
10    int k = 4;
11
12    cout << stirling(n, k);
13
14    return 0;
```

```
1
```





JĘZYK PYTHON

Twoje zadania

1. Program w języku Python na standardowym wyjściu wypisuje wartość liczby Stirlinga II rodzaju.

```
1 def stirling(n, k):  
2     pass  
3  
4 n = 7  
5 k = 4  
6 print(stirling(n, k))
```

```
1
```



Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Rekurencja – zadania maturalne

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

I + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) objaśnia, a także porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przy tym przykłady problemów i algorytmów, w szczególności:

b) rekurencję (do generowania ciągów liczb, potęgowania, sortowania liczb, generowania fraktali),

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Prześledzisz sposób rozwiązywania zadań maturalnych wykorzystujących rekurencję.
- Rozwiążesz samodzielnie kilka zadań maturalnych, używając rekurencji.
- Przeanalizujesz, w jaki sposób uniknąć typowych błędów w tego typu zadaniach.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Rekurencja – zadania maturalne”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj” dotyczącymi programowania.

Faza wstępna:

1. Chętna lub wybrana osoba przedstawia najważniejsze informacje dotyczące rekurencji.
2. Nauczyciel wprowadza uczniów szczegółowo w temat lekcji i jej cele. Może posłużyć się wyświetloną na tablicy zawartością sekcji „Wprowadzenie”.
3. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji oraz w odniesieniu do poznanych języków programowania.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel wyświetla sekcję „Przeczytaj”. Chętna lub wybrana osoba przedstawia najważniejsze informacje, jakie wyciągnęła z analizy rozwiązań wykonanej przed lekcją. Nauczyciel odpowiada na ew. pytania.
2. **Praca z multimediami.** Uczniowie analizują problem 1 z sekcji „Prezentacja multimedialna”. Indywidualnie go rozwiązują dla funkcji `sklej(7)`, a następnie porównują swoje rozwiązanie z przedstawionym w prezentacji. W parach wymieniają się swoimi rozwiązaniami i analizują je.
3. **Ćwiczenia umiejętności.** Uczniowie pracując w parach rozwiązują problem 1 z sekcji „Sprawdź się” (w zależności od języka, w którym programują, rozwiązują ćwiczenia 5, 6 lub 7). Następnie porównują swoje rozwiązania z inną grupą, pracującą w tym samym języku. W kolejnym kroku na forum klasy omówione zostają ewentualne problemy i wątpliwości.

Faza podsumowująca:

1. Na koniec zajęć nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie rozwiązują ćwiczenia 1-4 z sekcji „Sprawdź się”.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.
- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Nauczyciel może zlecić przygotowanie quizu związanego z problemem rekurencji w celu przygotowania się do kartkówki lub sprawdzianu wiadomości.