



Zagadka Wież Hanoi w języku C++

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Symulacja interaktywna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Zagadka Wież Hanoi w języku C++

Źródło: Thuan Pham, domena publiczna.

Łamigłówka Wież Hanoi jest problemem, który polega na przeniesieniu danej liczby krążków – o różnych średnicach – z jednego słupka na drugi tak, aby docelowo były ułożone w kolejności od największego do najmniejszego. Zagadka ta oparta jest na legendzie o buddyjskich mnichach, którzy pojedynczo mają przenieść 64 krążki, czyli w celu rozwiązania problemu muszą wykonać aż $2^{64} - 1$ ruchów. W e-materiale [Zagadka Wież Hanoi](#) przedstawiliśmy najważniejsze informacje dotyczące tego zagadnienia.

W tym e-materiale poznasz implementację algorytmu rozwiązywania problemu Wież Hanoi w języku C++.

Więcej przykładów, ćwiczeń i rozwiązań znajdziesz w e-materiale [Zagadka Wież Hanoi – zadania maturalne](#).

Ciekawi cię, jak wyglądają implementacje w innych językach programowania? Możesz się z nimi zapoznać w dwóch pozostałych e-materiałach z tej serii:

- [Zagadka Wież Hanoi w języku Java](#),
- [Zagadka Wież Hanoi w języku Python](#).

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.

Twoje cele

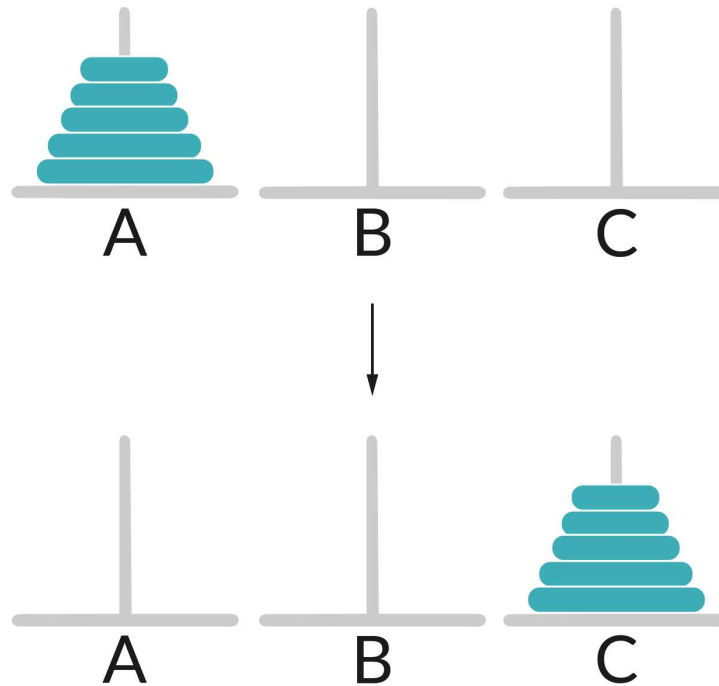
- Prześledzisz algorytm rozwiązania zagadki Wież Hanoi.
- Przeanalizujesz implementację algorytmu rozwiązania zagadki Wież Hanoi w języku C++.
- Rozwiążesz samodzielnie zadania związane z zagadką Wież Hanoi.

Przeczytaj

Przedstawienie problemu

Wiesz już, że [zagadka Wież Hanoi](#) polega na przeniesieniu krążków o różnych średnicach z jednego słupka na drugi. Warunkiem poprawnego rozwiązania zadania jest pojedyncze przenoszenie każdego krążka, który znajduje się na szczycie, a także brak możliwości położenia krążka o większej średnicy na krążku o mniejszej średnicy. Do dyspozycji mamy jednak pomocniczy słupek, na który możemy tymczasowo nakładać krążki.

Zatem mamy trzy słupki: A, B, C oraz n krążków. Początkowo wszystkie krążki położone są na słupku A. Docelowo wszystkie krążki mają znaleźć się na słupku C. Pomocniczym słupkiem jest słupek B. Celem naszego zadania jest przedstawienie ciągu ruchów, opisującego rozwiązanie zagadki Wież Hanoi.



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

Implementacja algorytmu w języku C++

Zagadkę Wież Hanoi dla n krążków rozwiążemy w sposób [rekurencyjny](#). Program przetestujemy dla trzech krążków.

Specyfikacja:

Dane:

- n – liczba krążków; liczba naturalna dodatnia
- A – nazwa słupka początkowego; znak

- B – nazwa słupka pomocniczego; znak
- C – nazwa słupka docelowego; znak

Wynik:

Program na standardowym wyjściu wypisuje ciąg ruchów opisujący rozwiązanie zagadki wież Hanoi dla n krążków, gdzie początkowo wszystkie znajdują się na słupku A. Na końcu wszystkie krążki mają znaleźć się na słupku C, a pomocniczym słupkiem jest B.

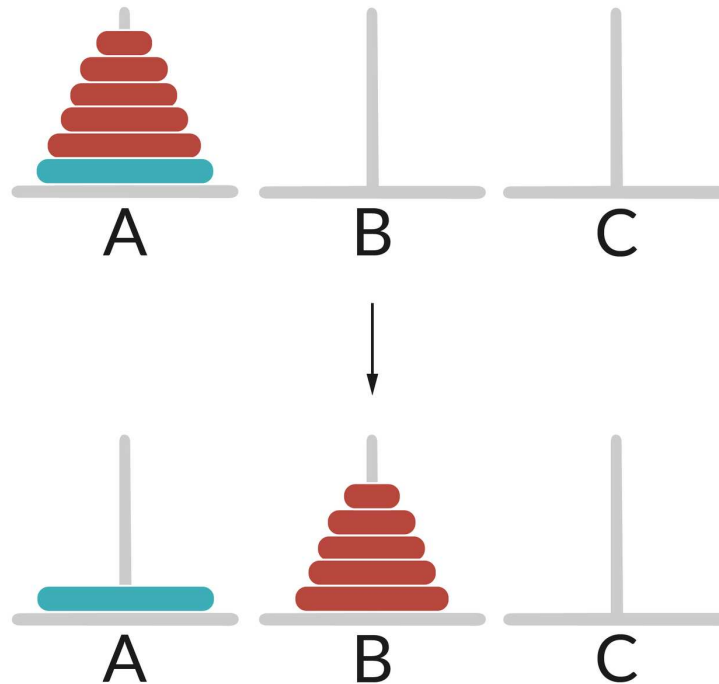
Tworzymy funkcję o nazwie `hanoi()`, której przyjmowanymi parametrami wejściowymi będzie n , czyli liczba krążków, a następnie A, B, C, czyli nazwy słupków: początkowego, pomocniczego, a także końcowego.

```
1 #include <iostream>
2 using namespace std;
3
4 void hanoi(int n, char A, char B, char C) {
5
6 }
```

Następnie, gdy liczba krążków jest większa od 0 (dla liczby krążków równej 0 zajdzie [przypadek bazowy](#)), możemy wykonywać algorytm. Wywołamy na początku rekurencyjnie funkcję `hanoi`, która zrealizuje przeniesienie $n - 1$ górnych krążków ze słupka A na słupkę B. Na słupku A pozostanie tylko jeden, n -ty, największy krążek. Reszta krążków będzie nałożona na słupkę B.

```
1 #include <iostream>
2 using namespace std;
3
4 void hanoi(int n, char A, char B, char C) {
5     if (n > 0) {
6         hanoi(n - 1, A, C, B);
7     }
8 }
```

Operację przeniesienia $n - 1$ górnych krążków ze słupka A na słupkę B przedstawia ilustracja:

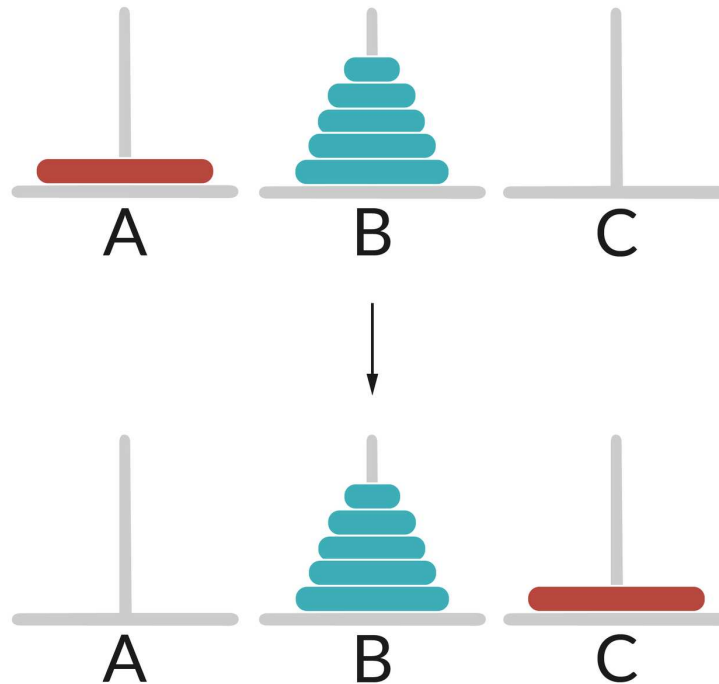


Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Największy krążek musi zostać przełożony na słupek docelowy C. Wypisujemy zatem na ekran komunikat: „Przeniesienie z” A „do” C.

```
1 #include <iostream>
2 using namespace std;
3
4 void hanoi(int n, char A, char B, char C) {
5     if (n > 0) {
6         hanoi(n - 1, A, C, B);
7         cout << "Przeniesienie z " << A << " do " << C << endl;
8     }
9 }
```

Operację przeniesienia n-tego krążka na słupek docelowy C, przedstawia rysunek:

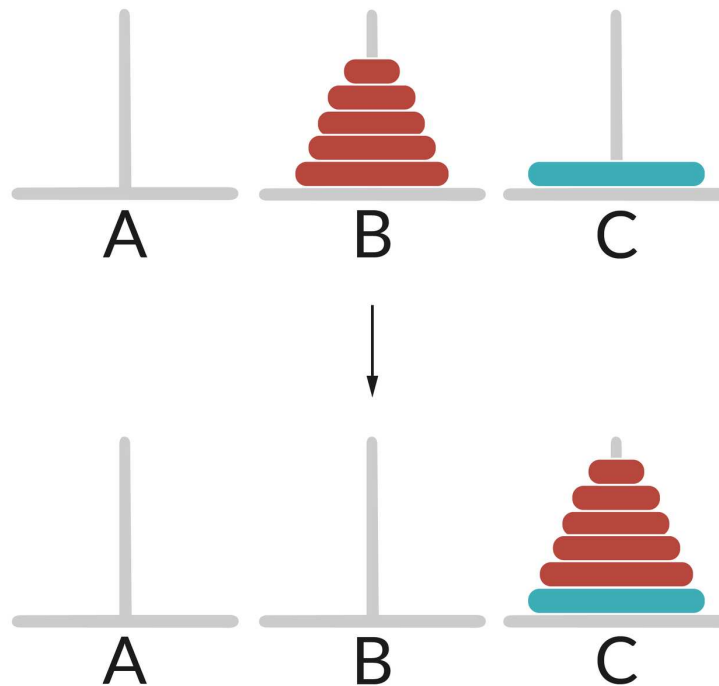


Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Na koniec musimy jeszcze wywołać rekurencyjnie funkcję `hanoi`, aby przenieść $n - 1$ krążków ze słupka B na słupek C, na którym położony jest już największy krążek.

Aby to zrealizować, pierwszym przyjmowanym argumentem będzie $n - 1$, ponieważ chcemy przenieść właśnie taką liczbę krążków. Drugim argumentem będzie B, ponieważ tym razem ten słupek jest słupkiem początkowym. Następnie umieszczamy A, ponieważ w tej sytuacji będzie on słupkiem pomocniczym, a na końcu słupek docelowy C.

```
1 #include <iostream>
2 using namespace std;
3
4 void hanoi(int n, char A, char B, char C) {
5     if (n > 0) {
6         hanoi(n - 1, A, C, B);
7         cout << "Przeniesienie z " << A << " do " << C << endl;
8         hanoi(n - 1, B, A, C);
9     }
10 }
```



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

Teraz wystarczy wywołać funkcję `hanoi`, np. dla trzech krążków z nazwami słupków: A, B, C.

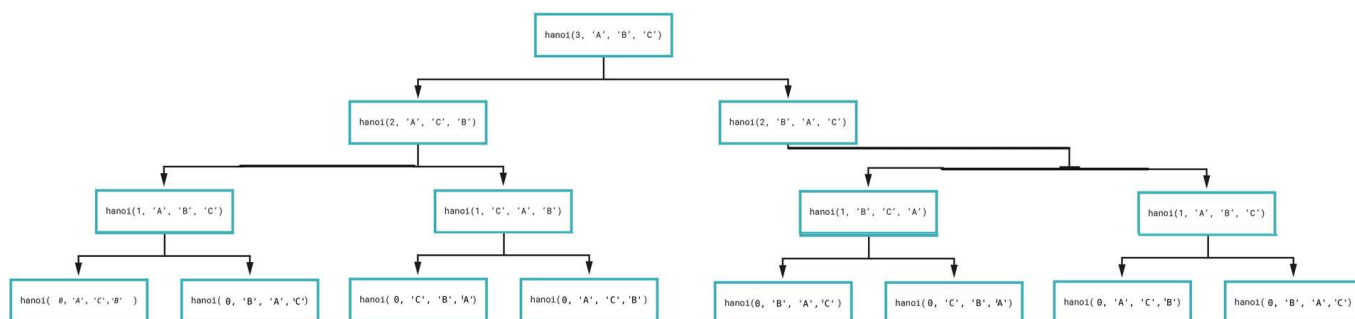
```
1 #include <iostream>
2 using namespace std;
3
4 void hanoi(int n, char A, char B, char C) {
5     if (n > 0) {
6         hanoi(n - 1, A, C, B);
7         cout << "Przeniesienie z " << A << " do " << C << endl;
8         hanoi(n - 1, B, A, C);
9     }
10 }
11
12 int main() {
13     hanoi(3, 'A', 'B', 'C');
14     return 0;
15 }
```

Ciąg ruchów otrzymany w wyniku wywołania funkcji dla trzech krążków będzie wyglądał następująco:

- Przeniesienie z A do C
- Przeniesienie z A do B
- Przeniesienie z C do B

- Przeniesienie z A do C
- Przeniesienie z B do A
- Przeniesienie z B do C
- Przeniesienie z A do C

Zobaczmy jeszcze, jak będzie wyglądało drzewo wywołań rekurencyjnych dla wywołania funkcji `hanoi(3, 'A', 'B', 'C')`. Dzięki niemu, będziemy mogli zobaczyć wszystkie wywołania funkcji `hanoi`. Lewa strzałka odchodząca od danej funkcji, będzie wskazywała na pierwsze wywołanie funkcji, umieszczone w szóstej linijce naszego kodu, natomiast prawa strzałka wskaże nam drugie wywołanie, z ósmej linijki. Aby odczytać rozwiązanie problemu z drzewa wywołań rekurencyjnych, najpierw musimy odczytać ruchy z lewego poddrzewa węzła, następnie ruch należący do węzła – ojca, a na końcu ruchy z prawego poddrzewa.



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

Zagadka wież Hanoi jest przykładem zadania, którego złożoność obliczeniowa wzrasta bardzo szybko wraz z liczbą krążków n . Przedstawiony algorytm ma złożoność **wykładniczą**, to znaczy $O(2^n)$.

Słownik

algorytm rekurencyjny

algorytm, w którym funkcja lub procedura wywołuje samą siebie; parametry wywołania takiej funkcji lub procedury rekurencyjnej mogą się zmieniać; odwołania rekurencyjnie pozostawiają na stosie systemowym adres i parametry powrotu do danej funkcji lub procedury

zagadka Wież Hanoi

problem, polegający na przeniesieniu wieży z krążków z jednego słupka na drugi, przy czym można przenosić tylko po jednym krążku z wierzchu stosu; nie jest dozwolone położenie większego krążka na mniejszym

przypadek bazowy

warunek w funkcji rekurencyjnej, po spełnieniu którego funkcja nie wywołuje samej siebie po raz kolejny

złożoność wykładnicza

złożoność, która jest określona za pomocą funkcji wykładniczej, rośnie wykładniczo w zależności od rozmiaru danych wejściowych n

Symulacja interaktywna

Polecenie 1

Przeanalizuj symulację interaktywną, wizualizującą kolejne kroki rozwiązania zagadki Wież Hanoi, dla wybranej liczby krążków, gdzie na początku wszystkie krążki znajdują się na słupku *A*, na końcu powinny znaleźć się na słupku *C*, a słupkiem pomocniczym jest słupek *B*.

Przeanalizuj kolejne kroki dla jednego, dwóch, trzech, a następnie dla czterech krążków. Zauważ, że rozwiązanie dla czterech krążków budujemy na podstawie rozwiązania dla trzech krążków, rozwiązanie dla trzech krążków budujemy na podstawie rozwiązania dla dwóch krążków itd.

Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/DVctv8oxJ>

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

Polecenie 2

Przygotuj notatkę ze swoimi spostrzeżeniami dotyczącymi symulacji.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Napisz program wyświetlający liczbę ruchów niezbędnych do rozwiązania zagadki wież Hanoi dla k krążków. Twój program powinien wypisać na ekranie liczbę wykonanych ruchów. Przetestuj swój program dla 7 krążków.

Specyfikacja problemu:

Dane:

- k – liczba krążków; liczba naturalna

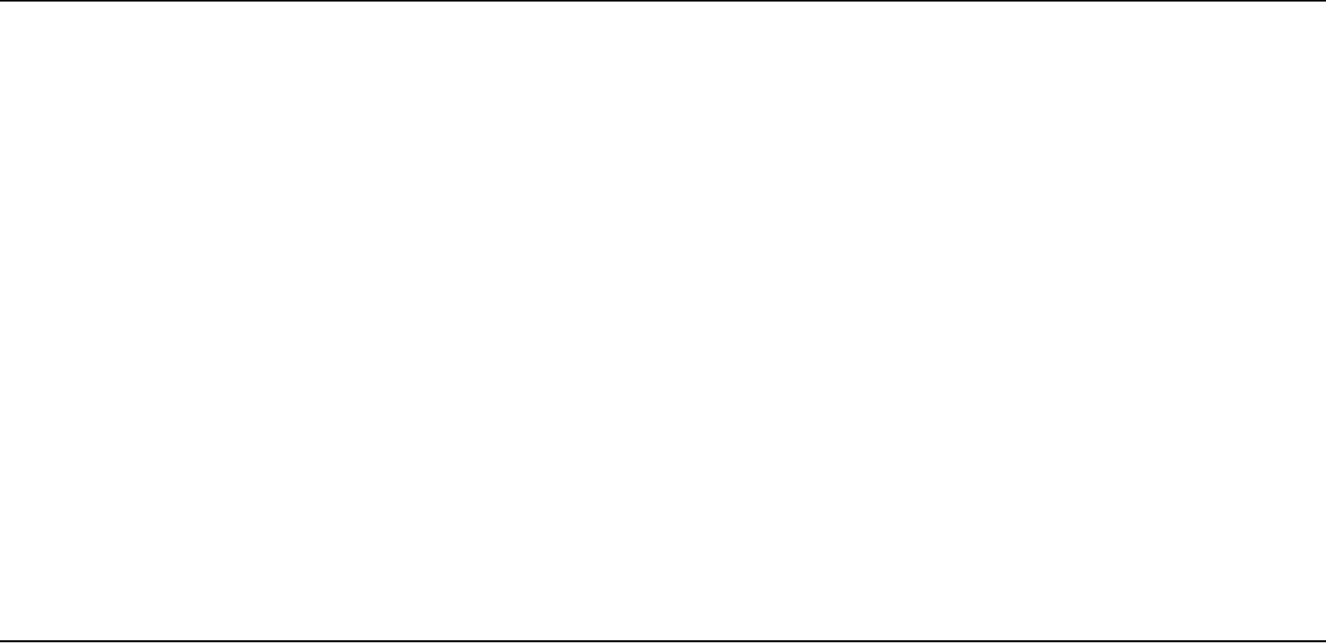
Wynik:

Program wypisuje liczbę ruchów, niezbędnych do rozwiązania zagadki wież Hanoi dla k krążków.

Twoje zadania

1. Program wyświetla liczbę ruchów opisujących rozwiązanie zagadki Wież Hanoi dla k krążków.

```
1 #include <iostream>
2 using namespace std;
3
4 int licznik;
5 int k;
6
7 void hanoi(int n, char A, char B, char C) {
8     // Tutaj dopisz kod
9 }
10
11 int main() {
12     licznik = 0;
13     k = 7;
14
```





Napisz program, który będzie wypisywał m -ty ruch, opisujący rozwiązanie Wieży Hanoi dla n krążków. Niech odpowiedź będzie w postaci: „Przeniesienie z X do Y”. Przetestuj swój program dla wieży składającej się z ośmiu krążków i wypisz piąty ruch.

Specyfikacja problemu:

Dane:

- n – liczba krążków; liczba naturalna
- m – numer ruchu, który ma zostać wypisany; liczba naturalna dodatnia, $m < 2^n$

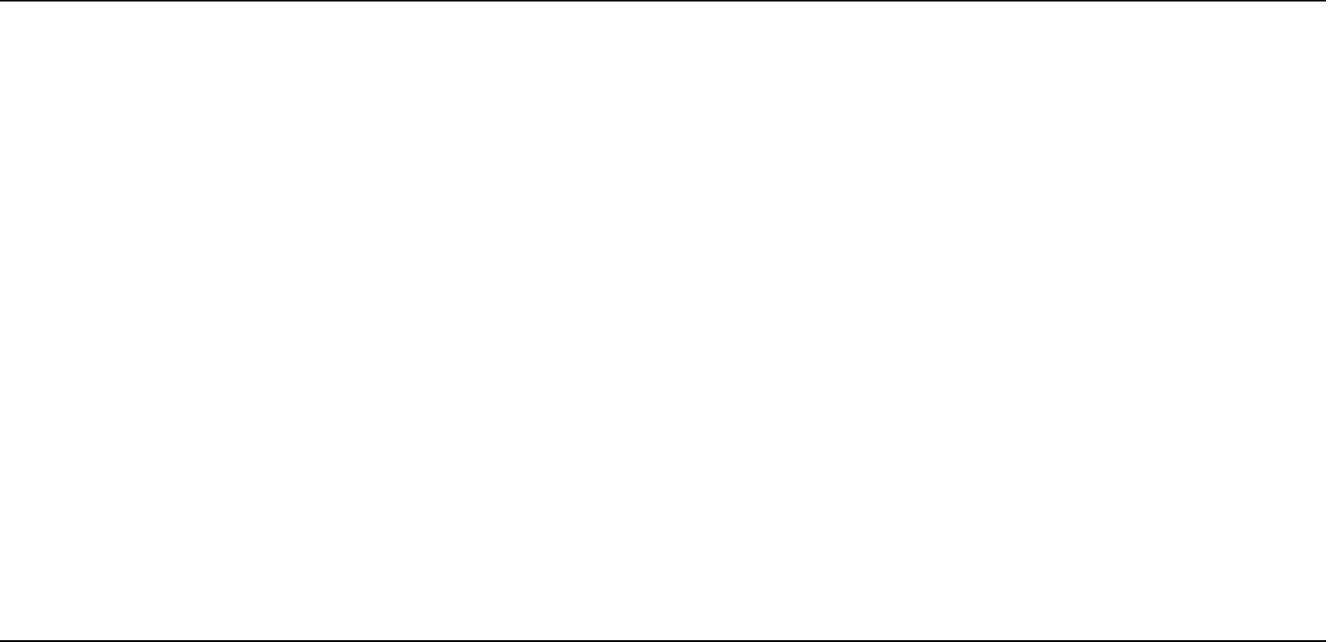
Wynik:

Program wyświetla m -ty ruch opisujący rozwiązanie zagadki Wieży Hanoi dla n krążków.

Twoje zadania

1. Program wyświetla opis m -tego ruchu, rozwiązującego zagadkę Wieży Hanoi dla n krążków.

```
1 #include <iostream>
2 using namespace std;
3 int licznik, n, m;
4
5 void hanoi(int n, char A, char B, char C) {
6     // Tutaj dopisz kod
7 }
8
9 int main() {
10     licznik = 0;
11     n = 8;
12     m = 5;
13     hanoi(n, 'A', 'B', 'C');
14     return 0;
```





Napisz program, który będzie wypisywał liczby krążków na poszczególnych słupkach, po wykonaniu m -tego ruchu opisującego rozwiązanie zagadki Wież Hanoi dla n krążków. Na początku wszystkie krążki powinny znajdować się na słupku początkowym. Jako pierwsza powinna zostać wypisana liczba krążków na słupku początkowym, następnie – w nowej linii – liczba krążków na słupku pomocniczym, a na końcu – w trzeciej linii – liczba krążków na słupku docelowym. Przetestuj swój program dla dziesiątego ruchu, dla wieży składającej się z ośmiu krążków.

Specyfikacja problemu:

Dane:

- n – liczba krążków; liczba naturalna
- m – numer ruchu, po którym ma zostać wypisana liczba krążków na słupkach; liczba naturalna dodatnia, $m < 2^n$

Wynik:

Program wypisuje liczby krążków na słupkach: początkowym, pomocniczym, docelowym, po wykonaniu m -tego ruchu opisującego rozwiązanie zagadki Wież Hanoi dla n krążków.

Twoje zadania

1. Program wyświetla liczbę krążków na słupkach: początkowym, pomocniczym, docelowym (każdy w nowej linii), po wykonaniu m -tego ruchu opisującego rozwiązanie zagadki Wież Hanoi dla n krążków.

```
1 #include <iostream>
2 using namespace std;
3 int licznik, n, m, krazkiNaA, krazkiNaB, krazkiNaC;
4
5 void hanoi(int n, char A, char B, char C) {
6     // Tutaj dopisz kod
7 }
8
```

```
9 int main() {  
10     licznik = 0;  
11     krazkiNaB = 0;  
12     krazkiNaC = 0;  
13     n = 8;  
14     krazkiNaA = n;
```

```
1
```

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Zagadka Wież Hanoi w języku C++

Grupa docelowa:

Liceum ogólnokształcące i technikum, 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

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 algorytm rozwiązania zagadki Wież Hanoi.
- Przeanalizujesz implementację algorytmu rozwiązania zagadki Wież Hanoi w języku C++.
- Rozwiążesz samodzielnie zadania związane z zagadką Wież Hanoi.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

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

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

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.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Zagadka Wież Hanoi w języku C++”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj” dotyczącymi programowania.

Faza wstępna:

1. Wyświetlenie przez nauczyciela tematu i celów zajęć, przejście do wspólnego ustalenia kryteriów sukcesu.
2. Rozpoznawanie wiedzy uczniów.

Faza realizacyjna:

1. Uczniowie analizują przykład z sekcji „Przeczytaj” i powtarzają zaprezentowane rozwiązanie na swoim komputerze.
2. **Praca z multimediami.** Uczniowie w zespołach dwuosobowych zapoznają się z treścią polecenia nr 1: „Przeanalizuj symulację interaktywną wizualizującą kolejne kroki rozwiązania zagadki Wież Hanoi dla wybranej liczby krążków. Spróbuj zapisać rozwiązanie tego zadania dla trzech krążków. Zastanów się, czy przedstawiona symulacja może ci znacząco pomóc w wykonaniu tego zadania? Ile kroków należy wykonać dla trzech, czterech i dla pięciu krążków?” z sekcji „Symulacja interaktywna” i wspólnie analizują kolejne kroki rozwiązania postawionego problemu.
3. **Ćwiczenie umiejętności.** Prowadzący zapowiada uczniom, że będą rozwiązywać ćwiczenie nr 1 z sekcji „Sprawdź się”. Uczniowie wykonują je w parach. Po ustalonym czasie następuje porównanie napisanych kodów podczas wspólnego omówienia rozwiązań.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie” i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązywania ćwiczeń z programowania w języku C++.

Praca domowa:

1. Uczniowie wykonują ćwiczenie 2 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.

Wskazówki metodyczne:

- Treści w sekcji „Symulacja interaktywna” można wykorzystać na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.