

Algorytm Euklidesa w języku C++

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



Algorytm Euklidesa w języku C++

Źródło: Alex Lehner, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Znamy już [Algorytm Euklidesa](#) w dwóch wariantach. Stosujemy go jako metodę wyznaczania największego wspólnego dzielnika oraz najmniejszej wspólnej wielokrotności dwóch liczb.

Największy wspólny dzielnik wykorzystujemy podczas skracania ułamków.

W tym e-materiale zaimplementujemy go w języku C++.

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

- [Algorytm Euklidesa w języku Java](#),
- [Algorytm Euklidesa w języku Python](#).

Więcej zadań? Sięgnij do [Algorytm Euklidesa – zadania maturalne](#).

Twoje cele

- Przeanalizujesz dwie wersje algorytmu Euklidesa.
- Wskażesz, jakie są różnice między zoptymalizowanym i niezoptymalizowanym algorytmem Euklidesa.
- Wykorzystasz algorytm Euklidesa do rozwiązywania problemów życia codziennego.

Przeczytaj

Implementacja algorytmu Euklidesa w języku C++

Problem 1

Zaimplementuj algorytm Euklidesa, wykorzystując:

- odejmowanie,
- resztę z dzielenia.

Specyfikacja problemu:

Dane:

- a, b – liczby naturalne dodatnie dla których program będzie wyznaczał NWD

Wynik:

Na standardowym wyjściu program wypisuje NWD liczb a i b.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     // tutaj dopisz kod
8 }
```

Polecenie 1

Porównaj swoje rozwiązanie z filmem.



Algorytm Euklidesa - zastosowanie różnych pętli

Implementacja algorytmu w języku C++



Film dostępny pod adresem </preview/resource/R12rgKYS7tEXF>

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

Plik o rozmiarze 611.00 B w języku polskim

Polecenie 2

Przeanalizuj prezentację. Porównaj ze sobą działanie dwóch wersji algorytmu dla różnych par liczb.

Mamy napisać program, który obliczy NWD dwóch podanych przez użytkownika liczb naturalnych dodatnich. Wprowadzona zostanie także trzecia wartość: wskaże ona typ algorytmu Euklidesa.

Trzecim parametrem jest:

- 1 - dla algorytmu Euklidesa z resztą z dzielenia,
- 2 - dla algorytmu Euklidesa z odejmowaniem.

2

Najpierw zadeklarujemy zmienne, w których przechowywać będziemy podane przez użytkownika liczby oraz wybrany typ algorytmu.

```
1 int a;  
2 int b;  
3 int typ;
```

Poprosimy użytkownika o wprowadzenie trzech liczb. Dla dwóch pierwszych obliczymy NWD, trzecia wskaże typ algorytmu Euklidesa. Musimy

1

3

pamiętać, że do pobrania danych z klawiatury potrzebna jest biblioteka `iostream`. Jeżeli jeszcze jej nie dołączyliśmy, zrobimy to na początku programu:

```
1 #include <iostream>
```

Dalej wpisujemy polecenia:

```
1 std::cout << "Proszę  
wprowadzić dwie liczby  
oraz typ: ";  
2 std::cin >> a >> b >> typ;
```

4

Następną czynnością jest napisanie instrukcji warunkowej `if`, która posłuży do wybrania odpowiedniego wariantu algorytmu Euklidesa:

```
1 if (typ == 1 || typ == 2)  
2 {  
3     if (typ == 1)  
4     {  
5     }  
6     if (typ == 2)  
7     {  
8     }  
9 }  
10 else std::cout << "Podany  
typ nie istnieje.";
```

Jeżeli użytkownik wprowadzi złą wartość, program wyświetli komunikat o błędzie.

Wewnątrz instrukcji warunkowej `if` dla wartości `typ == 1` utworzymy pętlę `while`. Jej działanie zakończy się w momencie, gdy zmienna `b` przyjmie wartość `0`:

5

```
1 while (b != 0)
2 {
3 }
```

6

W ciele pętli umieścimy zoptymalizowany algorytm Euklidesa. Najpierw zadeklarujemy pomocniczą zmienną `pom`, której przypiszemy wartość reszty z dzielenia `a` przez `b`. Następnie zmiennej `a` przypiszemy wartość zmiennej `b`, zaś w `b` zapiszemy wartość pobraną ze zmiennej `pom`.

```
1 int pom = a % b;
2 a = b;
3 b = pom;
```

Poza pętlą wypisujemy wartość zmiennej `a` - to w niej został zapisany największy wspólny dzielnik:

```
1 std::cout << "NWD to: " <<
  a;
```

7

8

Teraz w instrukcji `if` z warunkiem `typ == 2` dodamy pętlę `while`, która będzie wykonywana, dopóki zmienne `a` i `b` nie są sobie równe.

```
1 while (a != b)
2 {
3 }
```


9

Do środka pętli `while` wstawimy algorytm Euklidesa z odejmowaniem. Od większej liczby będziemy odejmować mniejszą:

```
1 if (a > b) a -= b;  
2 else b -= a;
```

10

Obie zmienne mają po zakończeniu działania pętli tę samą wartość. Jako NWD możemy wypisać dowolną z nich.

```
1 std::cout << "NWD to: " <<  
  a;
```

11

Skończyliśmy pisać program.

Dodatkowo w finalnej wersji kodu zdefiniowana została zmienna `licznik`, której zadaniem będzie zliczanie liczby wykonań pętli dla wskazanej przez użytkownika wersji algorytmu. Oto pełny kod:

```
1 #include <iostream>  
2  
3 int main ()  
4 {  
5     int a;  
6     int b;  
7     int typ;  
8     int licznik = 0;  
9     std::cout << "Proszę  
  wprowadzić dwie liczby
```

```

oraz typ: ";
10     std::cin >> a >> b >>
typ;
11     if (typ == 1 || typ ==
2)
12     {
13         if (typ == 1)
14         {
15             while (b != 0)
16             {
17                 int pom = a %
b;
18                 a = b;
19                 b = pom;
20                 licznik++;
21             }
22             std::cout << "NWD
to: " << a;
23         }
24         if (typ == 2)
25         {
26             while (a != b)
27             {
28                 if (a > b) a -
= b;
29                 else b -= a;
30                 licznik++;
31             }
32             std::cout << "NWD
to: " << a;
33         }
34     }
35     else std::cout <<
"Podany typ nie
istnieje.";
36     std::cout << "Liczba
wykonan petli: " <<
licznik << std::endl;
37 }

```

Dla przejrzystości, algorytm może zostać zapisany również w postaci funkcji. W poniższym kodzie za algorytm z resztą z dzielenia odpowiada funkcja NWD1, a za algorytm z odejmowaniem – NWD2.

```
1 #include <iostream>
2
3 int NWD1(int a, int b)
4 {
5     while (b != 0)
6     {
7         int pom = a % b;
8         a = b;
9         b = pom;
10    }
11    return a;
12 }
13
14 int NWD2(int a, int b)
15 {
16     while (a != b)
17     {
18         if (a > b) a -= b;
19         else b -= a;
20     }
21     return a;
22 }
23
24 int main ()
25 {
26     int a;
27     int b;
28     int typ;
29     std::cout << "Proszę
30     wprowadzić dwie liczby
31     oraz typ: ";
32     std::cin >> a >> b >>
33     typ;
34     if (typ == 1 || typ ==
35     2)
36     {
37         if (typ == 1)
38         {
```

```
35         std::cout << "NWD
to: " << NWD1(a, b);
36     }
37     if (typ == 2)
38     {
39         std::cout << "NWD
to: " << NWD2(a, b);
40     }
41 }
42 else std::cout <<
"Podany typ nie
istnieje.";
43 }
44
```

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

Podsumowanie

Problem 2

Wyznamy **NWD** liczb 35 i 14 przy użyciu dwóch wariantów algorytmu Euklidesa.

Algorytm Euklidesa z odejmowaniem

W tej wersji algorytmu interesuje nas różnica liczb, których NWD chcemy znaleźć. Odejmujemy zawsze mniejszą liczbę od większej.

Szukając NWD liczb 35 i 14, wykonamy następujące czynności:

1. Odejmemy 14 od 35 ($35 > 14$) – powstaje nowa para liczb: 21 i 14.
2. Odejmemy 14 od 21 – otrzymujemy następną parę: 7 i 14.
3. Odejmemy 7 od 14 – pozostaje nam para liczb 7 i 7.

Cykl odejmowania i tworzenia nowych par kończymy w chwili, gdy otrzymamy dwie identyczne liczby; każda z nich to NWD. W przypadku 35 i 14 jest to 7.

Implementacja w języku C++

```
1 int a = 35;
2 int b = 14;
```

```
3
4 while (a != b)
5 {
6     if (a > b)
7     {
8         a -= b;
9     }
10    else
11    {
12        b -= a;
13    }
14
15 }
```

Algorytm Euklidesa z resztą z dzielenia

Wykonujemy następujące działania:

1. Obliczamy resztę z dzielenia 35 przez 14. Wynosi ona 7.
2. Obliczamy resztę z dzielenia 14 przez 7; wynik operacji to 0. Pozostały nam liczby 7 i 0.

Ponieważ jedna z liczb ma wartość zero, druga z nich (czyli 7) jest największym wspólnym dzielnikiem.

Implementacja algorytmu w języku C++

```
1 int a = 35;
2 int b = 14;
3
4 while (b != 0)
5 {
6     int pom = a % b;
7     a = b;
8     b = pom;
9 }
```

Słownik

NWD

akronim pojęcia „największy wspólny dzielnik”

Audiobook

Polecenie 1

Zapoznaj się z audiobookiem i wykonaj polecenie 2.

Audiobook można wysłuchać pod adresem: <https://zpe.gov.pl/b/PRand33dv>

Imię Euklidesa związało się na zawsze z jedną z gałęzi geometrii – zwanej geometrią euklidesową. Tak trwały pomnik zdobył on zasłużenie dzięki słynnej swej pracy „Elementy”. Przez kilkanaście wieków na całym świecie uczono geometrii w szkołach według „Elementów” Euklidesa. Nawet do dziś w szkołach angielskich podręczniki geometrii przypominają swoim opracowaniem jego dzieło. „Elementy” – to jedno z najbardziej popularnych i rozpowszechnionych dzieł w literaturze światowej. Mimo tak dużej popularności Euklidesa jako autora „Elementów” sama postać jego jest mało znana. Historia nie przekazała żadnych pewnych wiadomości o jego życiu, nawet dokładna data urodzin i śmierci nie jest znana. Przypuszcza się (...), że okres działalności Euklidesa przypada na lata panowania Ptolomeusza Sotera I (...). Za rządów tego władcy stolica Aleksandria stała się centrum życia naukowego i kulturalnego, ściągającym wielu wybitnych naukowców z różnych stron świata, między innymi z Grecji. Słynna ówczesnie Szkoła Aleksandryjska skupiała wielu matematyków, wśród których Euklides był jednym z pierwszych jej wykładowców. Jak wskazują zachowane po nim dzieła, był to nauczyciel o wyjątkowym talencie dydaktycznym. Przypuszcza się, że Euklides był wychowankiem Akademii Platońskiej, gdzie posiadał głęboką wiedzę mając dostęp do najlepszych prac matematyków i filozofów greckich. Dzieła jego noszą na sobie piętno filozofii platońskiej, na przykład z dużą skrupulatnością omija wszelkie problemy związane z praktyką. Pewne światło na postać Euklidesa – człowieka, matematyka, filozofa – rzucają dwie anegdoty, których autentyczność, jak każdej anegdoty, jest oczywiście wątpliwa. Według jednej z nich król Ptolomeusz I przeglądając „Elementy” zapytał autora z nadzieją w głosie, czy nie ma krótszych dróg wiodących do geometrii, na co Euklides odrzekł: „W geometrii nie ma specjalnych dróg nawet dla królów”. Inna anegdota mówi, że pewien uczeń studiujący geometrię u mistrza Euklidesa po zapoznaniu się z pierwszym twierdzeniem zapytał, co mu przyjdzie z uczenia się geometrii. Na to Euklides przywołał niewolnika i powiedział: „Daj mu obola, ponieważ musi on mieć zysk ze wszystkiego, czego się uczy”. Matematyk Pappus (...) wychwalała niezwykłą uczciwość, wyjątkową skromność, łagodność i jednocześnie niezależność, jakie cechowały Euklidesa.

Euklides był bardzo płodnym autorem. Wiadomo, że napisał co najmniej 10 traktatów, wśród których „Elementy”, 13 składające się z trzynastu ksiąg, uchodzą za największe wydarzenie w historii matematyki. Jest to pierwsze zachowane dzieło matematyczne, w którym metoda dedukcyjna została w pełni przedstawiona. W pracy tej, mającej charakter podręcznika, Euklides zawarł całą wiedzę matematyczną swoich poprzedników. Nie był więc samodzielny twórcą jej treści, poza małymi wyjątkami, jak przekroje stożkowe, geometria sferyczna. Dał się tu poznać jako doskonały systematyzator i najwybitniejszy dydaktyk, jakiego zna historia matematyki. Dzieło to powstało około 300 roku przed naszą erą, lecz najstarsze zachowane rękopisy z tekstem greckim pochodzą dopiero z X wieku naszej ery. Zachowało się jedynie kilka skrawków greckich papirusów z I wieku naszej ery. Mimo braku oryginału dzięki żmudnej pracy naukowców, drogą porównań różnych zachowanych najstarszych rękopisów, dało się zupełnie wiarygodnie odtworzyć grecki tekst. (...) Wspaniała praca Euklidesa „Elementy” to dzieło, które miało fundamentalne znaczenie przez z górą 2000 lat.

Źródło: Włodzimierz Krywicki, *Poczet wielkich matematyków*, Warszawa 1965, s. 12–16.

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

Polecenie 2




Poszukaj informacji na temat kolejnych ksiąg wchodzących w skład „Elementów”. W której z nich znajduje się omawiany algorytm?

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

Polecenie 3

Zapisz notatkę, w której podsumujesz najważniejsze informacje przedstawione w audiobooku.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Napisz program, który dla trzech liczb naturalnych: a, b, c wypisuje dwie z nich, które mają największy wspólny dzielnik.

Specyfikacja problemu:

Dane:

- a, b, c – liczby naturalne dodatnie

Wynik:

Na standardowym wyjściu program wypisuje dwie spośród trzech liczb, mające maksymalny NWD.

Jeśli wszystkie pary mają taki sam NWD, program wypisuje komunikat:

```
1 Pary mają taki sam NWD
```

Przykładowe wyjście dla a=7, b=100, c=150:

```
1 100 i 150
```

Twoje zadania

1. Program wypisuje dwie spośród trzech liczb, mające maksymalny NWD.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main () {
6     int a = 51;
7     int b = 34;
8     int c = 45;
```

```
9 // Tutaj dodaj kod.  
10 // Liczby wypisz zgodnie ze schematem wskazanym w zadaniu.  
11 }
```

```
1
```

Ćwiczenie 2



Firma *Warzywa Inc* produkuje włoszczyznę. Proces produkcji polega na umieszczeniu na plastikowej tacce marchewek, pietruszek, porów i selerów, owinięcie ich folią spożywczą i zapakowaniu w kartony zbiorcze. Firmie udało się zakontraktować u rolników a sztuk marchewek, b sztuk pietruszki, c sztuk porów i d sztuk selerów. Ile jednakowych paczek włoszczyzny może przygotować do sprzedaży firma *Warzywa Inc*, tak aby było ich jak najwięcej?

Prezes firmy, Jan Bajtek nie lubi marnowania żywności – zawsze zamawia warzywa w taki sposób, aby produkcja odbywała się bez strat i wykorzystane były **wszystkie** warzywa.

Swoje rozwiązanie przetestuj dla $a = 210$, $b = 150$, $c = 90$, $d = 60$.

Specyfikacja problemu:

Dane:

- a – liczba marchewek; liczba naturalna dodatnia
- b – liczba pietruszek; liczba naturalna dodatnia
- c – liczba porów; liczba naturalna dodatnia
- d – liczba selerów; liczba naturalna dodatnia

Wynik:

Na standardowym wyjściu program wyświetla liczbę paczek włoszczyzny.

Twoje zadania

1. Program wypisuje liczbę paczek włoszczyzny, które uda się wyprodukować z zadanej liczby warzyw.

```
1 #include <iostream>
```

```
2
3 using namespace std;
4
5 int main () {
6     int a = 210, b = 150, c = 90, d = 60;
7     // Wpisz właściwy kod.
8 }
```

```
1
```

Ćwiczenie 3

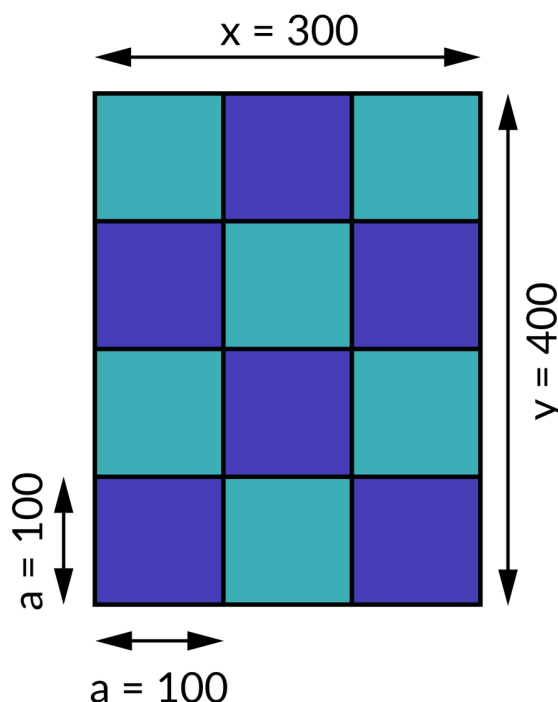


Bajtazar chce wyremontować posadzkę w swojej kuchni o wymiarach x na y . W tym celu chce kupić kwadratowe kafelki o boku długości a (jednostki są ustandaryzowane). Bajtazar chce kupić możliwie jak najmniej kafelek – zakładamy, że w sklepie będzie mógł dobrać dowolny rozmiar, który idealnie wpasuje się w podłogę kuchni (nie będzie konieczności docinania ani kupowania na zapas). Bajtazar nie lubi monotoni – kafelki będą w dwóch kolorach: morskim oraz fioletowym, ułożone naprzemiennie.

Pomóż Bajtazarowi przy remoncie: policz jaki wymiar a powinny mieć kafelki oraz ile powinien ich kupić w każdym z kolorów. Bajtazar chciałby rozpocząć układanie kafelek od lewego górnego rogu kuchni. W tym miejscu wymarzył sobie morski kafelek.

Przykład:

Podłoga o wymiarach $x=300$ i $y=400$. Długość boku kafelka to $a=100$. Musi kupić 6 kafeleków morskich i 6 fioletowych.



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

Swoje rozwiązanie przetestuj dla x wynoszącego 640, y równego 400.

Specyfikacja problemu:

Dane:

- x, y – szerokość i długość podłogi; liczby naturalne dodatnie

Wynik:

Na standardowym wyjściu program wyświetla w kolejnych wierszach: długość boku kafelka (a), liczbę kafelków morskich oraz liczbę kafelków fioletowych.

Przykładowe rozwiązania:

Podłoga o wymiarach $x=300$ i $y=400$. Długość boku kafelka to $a=100$. Musi zakupić 6 kafelków morskich i 6 kafelków fioletowych.

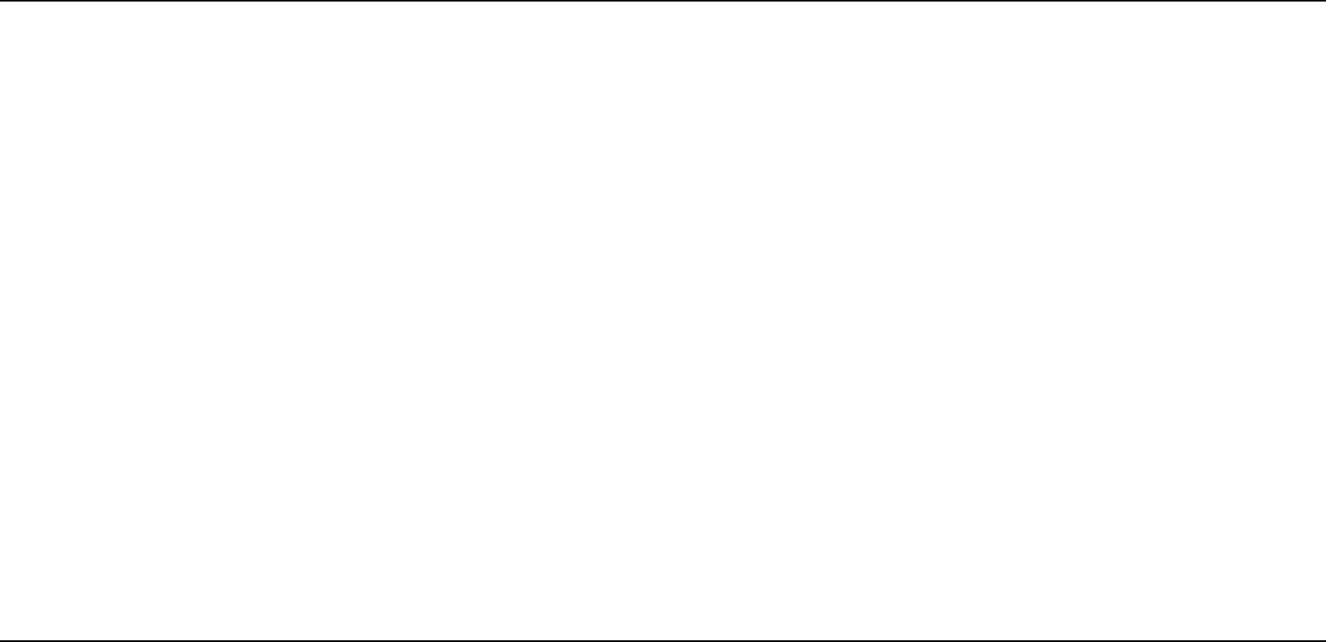
Podłoga o wymiarach $x=100$ i $y=100$. Długość boku kafelka to $a=100$. Musi zakupić 1 kafelek morski.

Twoje zadania

1. Program liczy długość pojedynczej kafelki oraz wskazuje liczbę kafelek fioletowych i morskich.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main () {
6     int x = 640;
7     int y = 400;
8 }
```

1



Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Algorytm Euklidesa 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. Rozumienie, analizowanie i rozwiązywanie problemów.

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

2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;

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:

1) zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach oraz algorytmy:

a) algorytm Euklidesa w wersji iteracyjnej i rekurencyjnej wraz z zastosowaniami,

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):

- Przeanalizujesz dwie wersje algorytmu Euklidesa.
- Wskażesz, jakie są różnice między zoptymalizowanym i niezoptymalizowanym algorytmem Euklidesa.
- Wykorzystasz algorytm Euklidesa do rozwiązywania problemów życia codziennego.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

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

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: „Algorytm Euklidesa w języku C++”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla i odczytuje temat lekcji oraz cele zajęć. Prosi uczniów o sformułowanie kryteriów sukcesu.
2. Rozpoznawanie wiedzy uczniów.

Faza realizacyjna:

1. Nauczyciel wyświetla zawartość sekcji „Przeczytaj”. Uczniowie zapoznają się z treścią problemu 1. W parach wypracowują rozwiązanie. Następnie porównują je z przedstawionym na filmie. W kolejnym kroku uczniowie wspólnie analizują prezentację, a następnie sprawdzają, o ile szybciej wykona się algorytm zoptymalizowany w porównaniu do niezoptymalizowanego dla różnych par liczb.
2. **Ćwiczenie umiejętności.** Poszukiwanie najefektywniejszego rozwiązania problemu. Uczniowie wykonują w parach ćwiczenie nr 1 z sekcji „Sprawdź się”, a następnie porównują swój kod omawiając go wspólnie na forum. Nauczyciel ocenia efektywność zastosowanego rozwiązania.
3. Liga zadaniowa – uczniowie pracując w parach, wykonują ćwiczenie nr 2: z sekcji „Sprawdź się”, a następnie dzielą się swoimi wynikami przez porównywanie napisanego kodu z inną grupą, która również zakończyła zadanie.

Faza podsumowująca:

1. Nauczyciel zadaje pytania podsumowujące, np.
 - w jakich przypadkach możemy użyć algorytm Euklidesa?
 - jakie są różnice między zoptymalizowanym i niezoptymalizowanym algorytmem Euklidesa?
2. Nauczyciel prosi uczniów o podsumowanie zgromadzonej wiedzy w zakresie programowania w języku C++.

Praca domowa:

1. Uczniowie wykonują ćwiczenie 3 z sekcji „Sprawdź się”.
2. Uczniowie zapoznają się z audiobookiem i wykonują polecenie 2 z sekcji „Audiobook”.

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:

- Uczniowie mogą wykorzystać multimedia z sekcji: „Przeczytaj”, „Audiobook”, „Sprawdź się” do przygotowania się do lekcji powtórkowej.