

Program koła informatycznego w zakresie algorytmiki i programowania, przygotowującego do startu w Olimpiadzie Informatycznej

Informacje o autorze:

- Imię i nazwisko autora: Ireneusz Bujnowski,
- Miejsce pracy: I Liceum Ogólnokształcące im. Adama Mickiewicza w Białymstoku
- Kontaktowy email: ibujnowski@gmail.com
- Telefon: 792 650 890

Krótki opis doświadczenia w pracy z uczniami uzdolnionymi informatycznie:

Jestem nauczycielem informatyki od 30 lat. Od ok. 16 lat pracuję z młodzieżą prowadząc zajęcia z programowania i algorytmiki przygotowując do startu w Olimpiadzie Informatycznej. Od 13 lat nieprzerwanie moi uczniowie uczestniczą w finale OI. W tym okresie zdobyli 27 tytułów laureata, 19 tytułów finalisty z wyróżnieniem oraz 39 tytułów finalisty Olimpiady Informatycznej.

W zawodach międzynarodowych zdobyli łącznie 19 medali: 4 medale w zawodach Międzynarodowej Olimpiady Informatycznej (1 złoty i 3 srebrne), 9 medali Olimpiady Krajów Bałtyckich (3 złote, 5 srebrnych i 1 brązowy), 6 medali Olimpiady Krajów Europy Środkowej (2 złote, 1 srebrny i 3 brązowe).

Główny cel kształcenia w ramach koła, opis głównych efektów uczenia się:

Celem koła jest przygotowanie się do startu w Olimpiadzie Informatycznej oraz lepsze przygotowanie do matury z informatyki. Z uwagi na różny poziom możliwości oraz pracowitości uczniów cel, którym jest start w OI, może być osiągnięty w ok. 50%. Zakładam też, że cała młodzież w klasach realizujących rozszerzony program informatyki powinna uczyć się programowania od początku klasy I. Efektem wczesnego kształcenia programowania i algorytmiki jest odsetek uczniów przystępujących do matury z informatyki. W mojej szkole jest to ok. 90%, co jest najwyższym wskaźnikiem w województwie podlaskim. Praca z młodzieżą na kołach i warsztatach w moim przypadku daje też inny efekt: prawie wszyscy absolwenci z klas informatycznych wybierają studia informatyczne.

Przedstawiony program nie jest skierowany do nauczycieli, którzy od lat mają finalistów Olimpiady Informatycznej, a raczej do tych nauczycieli którzy chcieliby uczyć programowania wchodząc razem z młodzieżą w trudną sferę algorytmiki.

Opis kompetencji uczniów przystępujących do programu i propozycja weryfikacji ich posiadania:

Proponuję nie wybierać uczniów do kół, każdy uczeń w z inną prędkością przełamuje trudności i w różnym czasie zaczyna programować. Znam przypadki, gdy uczeń w pierwszej klasie nie był w stanie przełamać się, a w trzeciej klasie zostawał finalistą OI. Program koła jest tak skonstruowany, by każdy mógł z sukcesem uczestniczyć co najmniej jeden rok. Daje to dobre podstawy zarówno dla „maturzystów” jak i „olimpijczyków”. Drugi rok, trudniejszy, generalnie przeznaczony jest dla osób najzdolniejszych, ale zachęcam do uczestnictwa wszystkich.

Czasem wsłuchanie się w trudniejsze zagadnienia pozwala z innej perspektywy spojrzeć na wcześniej realizowane tematy. Z założenia koło ma aktywizować młodzież, mobilizować do dodatkowego stałego wysiłku, co daje stały postęp dla wszystkich i umożliwia zdawanie informatyki na egzaminie maturalnym, gdzie ok. 40% punktów to algorytmika i programowanie.

Częstotliwość systematycznych zajęć, czas trwania zajęć:

Z mojego doświadczenia wynika, że w trakcie jednego roku szkolnego można przeprowadzić ok. 20 regularnych zajęć kół zainteresowań. W swoim projekcie zakładam więc pracę w cyklu 2-letnim (lub pracę na 2 poziomach zaawansowania). W każdym roku 20 obowiązkowych zajęć z wykładami realizowanych w 2-godzinnych blokach (zajęcia te powinny wyczerpać wszystkie zagadnienia z Sylabusu oraz pięć 4-godzinnych warsztatów w roku, które poświęcone mogą być na wyrównywanie poziomów uczniów, na wspólne rozwiązywanie zadań (to na początku np. po miesiącu zajęć), czy na zespołowe zawody programistyczne (np. przy okazji Nowego Roku, Dnia Wiosny, święta Szkoły, czy Dnia Dziecka). Daje to w sumie 60 godzin zajęć na każdym poziomie zaawansowania.

W mojej szkole dwa razy w roku odbywają się tygodniowe obozy naukowe oraz 4-godzinne warsztaty-zawody w każdą sobotę na kilku poziomach zaawansowania. Efekt współzawodnictwa i towarzyszących zawodom emocji są bardzo motywujące dla nauczycieli i uczniów. Zdaję sobie jednak sprawę z tego, że na początku bez widocznych efektów zarówno nauczyciele, jak i uczniowie nie poświęcą aż tyle dodatkowego czasu.

Warunki w tym infrastruktura niezbędne do realizacji zajęć (sprzęt, oprogramowanie, zasoby internetowe):

Prowadzenie koła nie wymaga specjalnego sprzętu, czy oprogramowania. Wystarczy dostęp do serwera, na którym nauczyciel ma uprawnienia wyższe niż uczniowie pozwalające na dodawanie zadań, tworzenie zawodów ew. oglądanie i porównywanie kodów uczniów. Jest kilka takich sprawdzarek (o których wiem), które udostępniają takie funkcje. Dobrym pomysłem, choć nie jest to niezbędne, jest posiadanie własnej sprawdzarki. Jest to kłopotliwe w utrzymaniu i nieustannym dostosowywaniu do zmieniającego się oprogramowania serwerowego. Własna sprawdzarka pozwala z czasem dostosować ją do naszych indywidualnych potrzeb i niejako przy okazji zbudować sobie zbiory problemów, zadań i materiałów dopasowanych do takiego serwisu.

Koncepcja i opis innych form kształcenia

(praca samodzielna, obozy, grywalizacja itp.):

Uzupełnieniem części stacjonarnej (wykładowej i warsztatowej) stanowią dwa tygodniowe obozy naukowe. Pierwszy z nich planowany jest na przełomie września i października – bezpośrednio przed pojawieniem się zadań I etapu Olimpiady Informatycznej. Dla klas I jest to sposób na wyrównanie umiejętności, ponieważ część uczniów uczyło się elementów programowania w szkole podstawowej, a także miejsce, gdzie bardzo pomaga się uczniom mającym problemy z początkowymi umiejętnościami związanymi z programowaniem. Dla klas II i III obóz jest miejscem integracji przed olimpiadą oraz początkiem ciężkiej pracy po „rozleniwiających” wakacjach. Podczas obozów dobrze sprawdzają się tzw. konsultanci – to uczniowie klas starszych, którzy służą pomocą uczniom klas młodszych. Są konsultanci z klas III

i ich zadaniem jest pomoc dla uczniów klas II, są też wybrani konsultanci z klas II, którzy z kolei pomagają uczniom klas I.

Drugi obóz organizowany jest przed II etapem OI. Zwykle na przełomie stycznia i lutego. Zwykle też od tego obozu pojawiają się zadania o wiele trudniejsze niż te, które rozwiązywali do tej pory.

Tematyka spotkań – grupa podstawowa i zaawansowana:

Proponowane tematy spotkań koła poziom podstawowy (lub pierwszy rok koła)

1. Zapoznanie ze środowiskiem CodeBlocks. Pierwsze programy w C++.
2. Instrukcje warunkowe oraz wyboru. Testowanie podzielności. Arytmetyka modulo.
3. Instrukcje pętli while oraz for.
4. Tablice jednowymiarowe – deklaracje, liniowe wyszukiwanie informacji w tablicach.
5. Tablice jednowymiarowe – Zliczanie, sumy prefiksowe – jako „pytania o przedziały”
6. Instrukcje „pętli w pętli” – sposoby rozwiązywania zadań z wieloma przypadkami w pojedynczym teście.
7. Tablice jednowymiarowe – sortowanie tablic, elementy złożoności obliczeniowej
8. Zmienne typu string i char – przetwarzanie tekstów.
9. Własna arytmetyka, implementacja wielkich liczb.
10. Funkcje w C++. Parametry funkcji oraz zasięg zmiennych – zmienne lokalne i globalne. Rekurencja, funkcje rekurencyjne.
11. Wyszukiwanie binarne – optymalne wyszukiwanie w tablicy z uporządkowanymi elementami. Biblioteka matematyczna (cmath)
12. Sortowanie z rekurencją, quicksort, mergesort. Metoda „dziel i zwyciężaj”.
13. Algorytm sortowania, funkcja sort z biblioteki algorithm, własna funkcja porównująca sortowane elementy.
14. Wyszukiwanie k-tego elementu w ciągu, najdłuższy podciąg rosnący.
15. Struktury danych w STL – pair, stack, queue, vector.
16. Przegląd algorytmów o różnej złożoności – elementy złożoności obliczeniowej. Koszt amortyzowany, gąsienica - technika dwóch wskaźników.
17. Testy pierwszości, sito Eratostenesa.
18. Programowanie zachłanne i programowanie dynamiczne. Wybrane algorytmy i problemy rozwiązywane dynamicznie
19. Proste algorytmy geometryczne na płaszczyźnie, wzajemne położenie punktu i prostej (odcinka), punktu i figury płaskiej, przecinanie się odcinków, pole powierzchni wielokątów, otoczka wypukła.
20. Elementy teorii gier – gry kombinatoryczne, gry bezstronne twierdzenie Sprague’a-Grundy’ego

Proponowane tematy spotkań koła poziom zaawansowany (lub drugi rok koła):

1. Drzewa binarne, przeglądanie drzew metodami PRE-, IN- oraz POST- order.
2. Kopiec binarny – sortowanie przez kopcowanie.
3. Drzewa binarne implementowane w tablicach – drzewo licznikowe, drzewo potęgowe.
4. Drzewo przedziałowe implementowane w tablicy.
5. Zaawansowane struktury danych w STL – set, multiset, mapa.
6. Zbiory rozłączne – FIND-UNION
7. Kolejka priorytetowa

8. Grafy – rodzaje i podstawowe własności, implementacja grafów. Przeglądanie grafu „wszerz”.
9. Grafy, przeglądanie grafu metodą „w głąb”, drzewo przeszukiwania DFS.
10. Spójność grafu, spójne składowe, sortowanie topologiczne.
11. Silna spójność grafów skierowanych – algorytmy Kosaraju oraz Tarjana, funkcja low, dwuspójność grafu.
12. Minimalne drzewa rozpinające – algorytm Kruskala.
13. Odległości w grafach I – BFS, algorytm Dijkstry,
14. Odległości w grafach II – algorytmy Belmana-Forda, SPFA,
15. Domknięcie przechodnie grafu, algorytm Floyda-Warshalla
16. Skojarzenia w grafach dwudzielnych.
17. LCA – najniższy wspólny przodek dwóch wierzchołków w drzewie – algorytm z wyszukiwaniem binarnym.
18. Range Minimum Query, algorytmy wyszukiwania wartości minimalnej w przedziale, zastosowanie RMQ do znajdowania LCA.
19. Przegląd algorytmów tekstowych – algorytm KMP, wyszukiwanie promieni palindromów w tekście.
20. Przegląd algorytmów tekstowych – rozwiązywanie wybranych problemów tekstowych przy pomocy hashowania.

Szczegółowy opis poszczególnych zajęć (20 godzin)

Zajęcia 1 (2 godziny)

Temat: Zapoznanie ze środowiskiem CodeBlocks. Pierwsze programy w C++.

Treści z sylabusu:

Matematyka:	Programowanie:	Algorytmika
Reprezentacja liczb w komputerze, zbiory	obsługa środowiska programistycznego, zmienne i typy zmiennych (całkowite i rzeczywiste), wejście i wyjście -formatowanie wyjścia, operatory arytmetyczne, działanie automatycznego systemu oceniającego	Algorytm liniowy

Czynności nauczyciela:

- prezentuje sposób zainstalowania środowiska do programowania w C++,
- opisuje strukturę programu w C++,
- omawia sposób wypisywania komunikatów tekstowych z poziomu C++,
- wyjaśnia typy danych oraz pojęcie zmiennej,
- pokazuje na ekranie (tablicy) sposoby wczytywania oraz wypisywania wartości liczbowych,
- opisuje sposób zapisu i działania operatorów arytmetycznych,
- demonstruje sposób działania systemu oceniającego rozwiązania uczniowskie.

Zadania do rozwiązania na lekcji:

- Wypisz na ekranie napis „Hello World!”,
- Wczytaj trzy liczby i wypisz je na ekranie w zmienionej kolejności
- Wczytaj dwie liczby całkowite. Wypisz wynik dodawania, odejmowania, mnożenia, dzielenia całkowitoliczbowego oraz reszty z dzielenia

Zadania domowe:

- Wczytaj liczbę n i wypisz sumę liczb $1..n$, na lekcji podana jest anegdota z Gausem i wzór sumy liczb $1..100$,
- Wczytaj dwie liczby całkowite r oraz h . Oblicz i wypisz objętość i pole powierzchni całkowitej oraz objętość stożka, walca i kuli.

Zadania na sprawdzarce:

Proste zadania sprawdzające umiejętność napisania bezbłędnego programu wypisującego teksty lub wartości liczbowe.

Uwagi do realizacji:

Zajęcia te są bardzo istotne. Młodzież nie może spotkać porażka zniechęcająca do udziału w dalszej części kursu programowania i algorytmiki, stąd ważne jest bardzo dokładne, wolne wyjaśnienie wszystkich problemów, które mogą zaistnieć przy instalacji, podczas pisania pierwszych programów, czy po pierwszych zgłoszeniach programów do systemu sprawdzającego.

Zajęcia 2 (2 godziny)

Temat: Instrukcje warunkowe oraz wyboru. Testowanie podzielności. Arytmetyka modulo.

Treści z sylabusu:

Matematyka:	Programowanie:	Algorytmika
podzielność i arytmetyka modulo, rachunek zdań	obsługa środowiska programistycznego, zmienne i typy zmiennych, instrukcja warunkowa, operatory i spójniki logiczne, instrukcja wyboru.	Algorytm z rozgałęzieniami

Zadanie sprawdzające opanowanie poprzedniej lekcji

- Dane są trzy liczby będące bokami trójkąta prostokątnego a , b , c , gdzie c jest długością przeciwprostokątnej. Wypisz obwód i pole powierzchni trójkąta.

Czynności nauczyciela:

- omawia instrukcję warunkową, w szczególności zapis *if (warunek) instrukcja*; oraz *if (warunek) instrukcja; else instrukcja2*;
- opisuje operatory i spójniki logiczne,
- omawia sposób zapisu instrukcji wyboru *switch*,
- wyjaśnia typ logiczny,
- opisuje jak wartości liczbowe traktowane są miejscach warunków,
- pokazuje na ekranie (tablicy) sposoby zapisu i działania instrukcji z warunkami i zmiennymi logicznymi,
- opisuje sposób zapisu i działania operatorów warunkowych,
- omawia własności arytmetyki modulo,

Zadania do rozwiązania na lekcji:

- Wczytaj liczbę całkowitą. Wypisz TAK, gdy jest parzysta, a NIE wpp,
- Wczytaj trzy liczby i wypisz największą z nich
- Wczytaj dwie daty, wypisz wcześniejszą,
- Wczytaj dwie liczby całkowite. Wypisz TAK, gdy jedna jest dzielnikiem drugiej, NIE wpp,

Zadania domowe:

- Dana jest liczba 0..10000. Podaj ile ma cyfr.
- Słoń w jednym kroku może przemieścić się o 1, 2, 3, 4 lub 5. Po ilu krokach trafi do przyjaciela odległego o dane n .

Zadania na sprawdzarce:

- $N!$ (N silnia) jest iloczynem liczb od 1 do N włącznie. Jaka jest ostatnia cyfra $N!$ dla danego N .
- Dane są dwa prostokąty. Dla każdego z nich dane są współrzędne przeciwległych wierzchołków. Oblicz pole wspólne tych figur.

Uwagi do realizacji:

W dalszym ciągu nauczyciel i uczniowie, którzy potrafią szybko wykonać zadanie pomagają uczniom, którzy z pisaniem programów spotykają się po raz pierwszy.

Zajęcia 3 (2 godziny)

Temat: Instrukcje pętli while oraz for.

Treści z sylabusu:

Matematyka:	Programowanie:	Algorytmika
NWD, algorytm Euklidesa, NWW, rozkład liczb na czynniki pierwsze	Instrukcje iteracyjne,	Algorytm Euklidesa (iteracyjny z odejmowaniem i dzieleniem), algorytm rozkładu liczb na czynniki pierwsze, zamiana liczb z systemu 10 na 2.

Zadanie sprawdzające opanowanie poprzedniej lekcji

- Dane są dwa pola na szachownicy $(w1, k1)$ oraz $(w2, k2)$, gdzie $(1 \leq w1, k1, w2, k2 \leq 8)$. Napisz program, który określi i wypisze w ilu minimalnie ruchach goniec z pola $(w1, k1)$ może dotrzeć do pola $(w2, k2)$ lub odpowiedź NIE, gdy nie jest to możliwe.

Czynności nauczyciela:

- omawia schemat działania instrukcji pętli while
- opisuje sposób zapisu warunku, przy którym wykonują się instrukcje objęte pętlą
- wyjaśnia pojęcie niezmiennika pętli
- pokazuje i analizuje działanie klasycznych algorytmów matematycznych zawierających pętlę while
- opisuje konstrukcję pętli for,
- analizuje działanie pętli for na przykładzie np. wyszukiwania dzielników liczby, czy naiwnego wyliczania sumy liczb z zakresu $\langle a, b \rangle$

Zadania do rozwiązania na lekcji:

- Ile dzielników ma wczytana liczba.
- Na ile czynników pierwszych można rozłożyć wczytaną liczbę.
- Dane są dwie liczby będące licznikiem i mianownikiem ułamka. Jaki będzie licznik i mianownik ułamka po maksymalnym skróceniu.

Zadania domowe:

- Dane są współrzędne dwóch punktów $(x1, y1)$ oraz $(x2, y2)$ będącymi końcami odcinka. Przez ile punktów kratowych przechodzi dany odcinek (liczymy razem z końcami odcinka).
- Czy dany rok jest rokiem przestępnym.

Zadania na sprawdzarce:

Koce – dane są dwie liczby oznaczające długości boków koca. Koc można złożyć na pół, gdy jego bok jest parzysty. Jaką minimalną powierzchnię można uzyskać składając koc.

Piłeczka – dana jest wysokość piłeczki i wysokość docelowa. Po każdym odbiciu piłeczka osiąga wysokość dwa razy większą niż poprzednio. Po ilu odbiciach osiągnie wysokość docelową.

Uwagi do realizacji:

Dobrze jest zilustrować działanie pętli while oraz for przy pomocy schematów blokowych. Choć jestem przeciwnikiem uczenia tego sposobu zapisu algorytmów, to jednak analiza działania pętli na schematach jest bardzo łatwa do pojęcia. Nie stwarza też trudności zapamiętanie działania wszystkich elementów pętli.

Zajęcia 4 (2 godziny)

Temat: Tablice jednowymiarowe – deklaracje, liniowe przetwarzanie informacji w tablicach

Treści z sylabusu:

Matematyka:	Programowanie:	Algorytmika
Ciągi arytmetyczne, geometryczne, wypełnianie tablicy elementami ciągu	Tablice statyczne jednowymiarowe, Testowanie programu przekierowanie strumienia w konsoli, wypisywanie pośrednich wyników.	Proste przetwarzanie tablic jednowymiarowych: wypełnianie, przeszukiwanie liniowe, znajdowanie min i max, odwracanie, przeszukiwanie cykliczne, przesunięcia elementów

Zadanie sprawdzające opanowanie poprzedniej lekcji

- Wyszukaj w internecie opis „Problem Colatza”. Dana jest liczba N. Po ilu minimalnie iteracjach liczba N będzie równa 1?
- Podaj sumę cyfr wczytanej liczby.

Czynności nauczyciela:

- omawia sposób deklaracji statycznych tablic jednowymiarowych
- opisuje operacje związane dostępu do elementów tablicy, nadawanie i zmiana wartości, odczyt elementów, zamiana elementów miejscami,
- pokazuje sposób testowania programów z tablicami z wykorzystaniem strumieni i systemowego schowka
- omawia wykorzystanie pętli for do liniowego przeszukiwania tablic,

Zadania do rozwiązania na lekcji:

- Wczytaj liczby do tablicy 20 elementowej i znajdź liczbę liczb, które są podzielne przez liczbę K wczytaną z klawiatury,
- Wczytaj liczby do tablicy 20 elementowej i podaj największą różnicę możliwą do uzyskania pomiędzy elementami nie będącymi sąsiadami w tablicy.

Zadania domowe:

- Wczytaj dane do tablicy. Podaj sumy elementów z pozycji parzystych i nieparzystych

Zadania na sprawdzarce:

Chrabąszcze – dane są pozycje początkowe chrabąszczy, liczby skoków i ich długości. Należy znaleźć najbliższą odległość chrabąszczy.

Bitfony – każdy uczeń ma nr telefonu tylko jednego innego ucznia. Zadanie polega na określeniu, czy uczeń A może przesłać informację (pośrednio bądź bezpośrednio) do ucznia B.

Uwagi do realizacji:

Lekcja bardzo ważna z punktu widzenia wielu kolejnych algorytmów. Należy zwrócić uwagę na „zaglądanie” do kolejnych komórek, na to jak działają zmienne sterujące. Należy zwrócić uwagę na możliwość wystąpienia błędów przy próbie odczytu poza tablicą. Należy też powiedzieć, że ważne jest miejsce w programie w którym tablica jest deklarowana, co skutkuje na przykład zerowaniem jej komórek lub ustalenie wartości losowych.

Zajęcia 5 (2 godziny)

Temat: Tablice jednowymiarowe – Zliczanie, sumy prefiksowe – jako „pytania o przedziały”.

Treści z sylabusu:

Matematyka:	Programowanie:	Algorytmika
Wyprowadzenie wzorów na n -ty wyraz ciągu arytmetycznego i geometrycznego, wzoru na sumę n elementów ciągu aryt. oraz geom. Dowody indukcyjne ww. wzorów	Tablice statyczne jednowymiarowe, Synchronizacja wejścia/wyjścia strumieniowego	Proste przetwarzanie tablic jednowymiarowych: cykliczne przesuwanie zawartości tablicy, sumy prefiksowe – proste „pytania o liczbę (sumę) elementów w przedziale”,

Zadanie sprawdzające opanowanie poprzedniej lekcji

- Dane są wysokości 20 budynków. Budynki o numerach parzystych znajdują się po jednej stronie ulicy, a nieparzyste po drugiej. Jaka jest największa suma wysokości trzech kolejnych stojących po jednej stronie ulicy budynków.

Czynności nauczyciela:

- omawia cykliczne przesuwanie zawartości tablicy przy pomocy odwracania fragmentów tablicy
- opisuje działanie sum prefiksowych oraz sposób ich wyliczenia
- wyjaśnia przykładowe zastosowania sum prefiksowych

Zadania do rozwiązania na lekcji:

- Dana jest waga n paczek. Mamy dwie ciężarówki – do pierwszej ładujemy paczki o numerach $1..k$, a do drugiej $k+1..n$. Jakie powinno być k , jeśli chcemy by różnica wag paczek ładowanych do ciężarówek była jak najmniejsza.
- Dany jest ciąg 0 oraz 1. Podaj ile zer jest od pozycji a do pozycji b w tym ciągu.
- Dany jest ciąg n elementów złożony z 0 i 1. Podaj długość najdłuższego spójnego podciągu 0.

Zadania domowe:

- Zliczanie liczby liczb parzystych i/lub nieparzystych w tablicy i odpowiedzi na przedziale

Zadania na sprawdzarce:

Ropucha – ropucha chce przejść przez rzekę po liściach które spadają z drzew. Mamy dany ciąg określający na jaką pozycję spadnie kolejny liść. Ropucha może przejść tylko wtedy, gdy na wszystkich pozycjach $1..n$ będzie liść. Pytanie brzmi: po upadku którego liścia ropucha będzie mogła przejść przez rzekę.

Domino – mamy danych n kostek domino. Należy określić, czy można ułożyć z tych kostek poprawny ciąg – taki, który nie ma rozgałęzień.

Uwagi do realizacji:

Ważny temat uświadamiający młodzieży jak istotny jest sposób wczytywania dużej liczby danych. Ważne jest również wprowadzenie do dowodzenia, by w przyszłości można było uzasadniać poprawność wymyślnego rozwiązania. Po kilku tematach z pętlami i tablicami młodzież nie powinna już mieć trudności z poprawnym zapisem pętli oraz przetwarzaniem tablic jednowymiarowych.

Zajęcia 6 (2 godziny)

Temat: Instrukcje „pętli w pętli” – sposoby rozwiązywania zadań z wieloma przypadkami w pojedynczym teście.

Treści z sylabusu:

Matematyka:	Programowanie:	Algorytmika
Sigma – zapis sumy, szeregi	Instrukcje iteracyjne, Tablica dwuwymiarowa	Niezmienniki pętli,

Zadanie sprawdzające opanowanie poprzedniej lekcji

- Dany jest przedział liczb całkowitych $\langle a, b \rangle$. Podaj ile liczb z tego przedziału dzieli się bez reszty przez k .

Czynności nauczyciela:

- omawia działanie zagłębionych pętli
- opisuje zmiany wartości zmiennych sterujących pętlami
- wyjaśnia możliwe błędy oraz sposoby grupowania instrukcji w pętlach
- pokazuje jak można wykorzystać zagłębione pętle do tworzenia rysunków ze znaków ASCII na ekranie
- omawia deklarację tablicy dwuwymiarowej oraz sposób odwoływania się do jej elementów

Zadania do rozwiązania na lekcji:

- Narysuj pełny kwadrat o boku k , złożony ze znaków '*' (gwiazdki).
- Narysuj pusty kwadrat o boku k , złożony ze znaków '*' (gwiazdki).
- Wypisz na ekranie tabliczkę mnożenia dla liczb 1..10.
- Oblicz sumę elementów leżących na dwóch przekątnych w tablicy kwadratowej

Zadania domowe:

- Napisz program, który „narysuje” na ekranie szachownicę złożoną ze znaków '*' (gwiazdka) i '.' (kropka). Wczytana z klawiatury liczba k określać powinna długość boku pojedynczego kwadratu tworzących szachownicę 8×8 .
- Narysuj na ekranie trójkąt prostokątny złożony ze znaków '*' o boku przyprostokątnych równych k .

Zadania na sprawdzarce:

- Kwadrat – przekątne tablicy podzieliły tablicę na 4 ćwiartki (przekątnych nie liczymy). Podaj sumy elementów w każdej utworzonej ćwiartce.
- Trójkąty liczb – w kolejnych wierszach znajdują się ciągi liczb zależnych od numeru wiersza i kolumny. Zadanie polega na odgadnięciu wzorów na liczby i odpowiedniej implementacji.

Uwagi do realizacji:

Problem implementacji zagłębionych pętli często przysparza trudności z deklaracją zmiennych sterujących pętlami, należy uczulić młodzież na powiązanie nazw zmiennych sterujących z rzeczywistą funkcją pełnioną w pętlach, np. w-wiersz, k-kolumna itp.

Zajęcia 7 (2 godziny)

Temat: Tablice jednowymiarowe – sortowanie tablic, elementy złożoności obliczeniowej

Treści z sylabusa:

Matematyka:	Programowanie:	Algorytmika
Porządek liniowy, relacje mniejszości i większości, pojęcie funkcji kwadratowej przy okazji złożoności	Tablice statyczne jednowymiarowe	Sortowanie przez wybieranie, sortowanie przez wstawianie, sortowanie przez zliczanie

Zadanie sprawdzające opanowanie poprzedniej lekcji

- Masz daną szachownicę 8x8, na której zapisano liczby. Oblicz i wypisz sumy liczb ze wszystkich pól czarnych i ze wszystkich pól białych.

Czynności nauczyciela:

- omawia i ilustruje działanie sortowania przez wybieranie
- opisuje zachowanie zmiennych sterujących pętlami
- razem z młodzieżą formułuje niezmienniki pętli
- pokazuje działanie sortowania przez wstawianie
- pokazuje technikę zliczania, opisuje działanie sortowania przez zliczanie,
- formułuje zalety i wady sortowania przez zliczanie

Zadania do rozwiązania na lekcji:

- Implementacja sortowania przez wybieranie
- Implementacja sortowania przez wstawianie
- Testy działania ww. algorytmów na różnej liczbie danych

Zadania domowe:

- Implementacja sortowania przez zliczanie
- Porównanie czasu działania algorytmów sortujących z lekcji i sortowania przez zliczanie

Zadania na sprawdzarce:

Czwórki - danych jest n różnych liczb całkowitych dodatnich. Ile jest wśród nich takich czwórek liczb (niekoniecznie różnych), że $x+y+z=w$.

Taśma - Dana jest taśma z n -liczbami. Jaka jest największa odległość pomiędzy dwiema różnymi wartościami na tej taśmie.

Uwagi do realizacji:

Podane sortowania co prawda nie mają wielkiego znaczenia w przypadku rozwiązywania problemów wymagających uporządkowania elementów, ale doskonale utrwalają zrozumienie zagłębionych pętli, umożliwiają przeprowadzenie prostej analizy złożoności, utrwalają sposoby wykonywania operacji na tablicach jednowymiarowych.

Zajęcia 8 (2 godziny)

Temat: Zmienne typu string i char – przetwarzanie tekstów.

Treści z sylabusu:

Matematyka:	Programowanie:	Algorytmika
Arytmetyka modulo	Łańcuchy znaków, typ char oraz string, rzutowanie typów,	Proste algorytmy dla tekstów (np. szyfry podstawieniowe), test palindromu, prosty test anagramów, wyszukiwanie wzorca w tekście metodą naiwną.

Zadanie sprawdzające opanowanie poprzedniej lekcji

- Dany jest zbiór n liczb całkowitych dodatnich określających długości odcinków. Sprawdź, czy można zbudować trójkąt wybierając dowolne trzy odcinki z tego zbioru.

Czynności nauczyciela:

- omawia typ char, jego zakres liczbowy oraz tablicę kodów ASCII
- opisuje „dualność” typu i sposób rzutowania zmiennej typu int na char i na odwrót
- wyjaśnia typ danych string, sposób odwołania się do poszczególnych znaków oraz niektóre funkcje jak np. napis.size() niezbędne przy iteracji
- pokazuje proste szyfry podstawieniowe (Cezara, rot13, atbash), test palindromu oraz sposób sprawdzenia, czy dwa teksty są anagramami
- opisuje naiwny sposób wyszukiwania wzorca w tekście

Zadania do rozwiązania na lekcji:

- Napisz program, który po wczytaniu tekstu wyświetli jego szyfrogram wykorzystując szyfr Cezara/rot13.
- Napisz program testujący, czy tekst będący zdaniem ze spacjami jest palindromem, czy też nie.

Zadania domowe:

- Zaimplementuj algorytm naiwny wyszukiwania wzorca.
- Napisz program, który po wczytaniu dwóch tekstów określi, czy są anagramami.

Zadania na sprawdzarce:

- Statystyka – spośród wszystkich komórek tablicy dwuwymiarowej znajdź wartość największą, najmniejszą oraz liczbę wartości parzystych i nieparzystych.
- Cyfry – masz danych n stringów zawierających litery i cyfry. Dla każdego łańcucha należy wypisać w osobnym wierszu cyfry z tego łańcucha.
- Stenotypistka – dany jest tekst s . Zamień wczytany tekst na zapis stenotypistki – dla każdego słowa z tekstu wypisz pierwszą i ostatnią literę, pomiędzy którymi jest znak ‘_’.

Uwagi do realizacji:

Temat wydaje się prosty w kontekście wcześniej przeciwiczonych tablic jednowymiarowych. Należy zwrócić uwagę, przypomnieć sposoby konwersji wartości pomiędzy różnymi typami. Będzie to przydatne na następnych zajęciach dotyczących wielkich liczb.

Zajęcia 9 (2 godziny)

Temat: Własna arytmetyka, implementacja wielkich liczb

Treści z sylabusu:

Matematyka:	Programowanie:	Algorytmika
Działania arytmetyczne w różnych systemach liczbowych	Rzutowanie typów, konwersja wartości pomiędzy różnymi typami danych	Arytmetyka wielkich liczb, schemat Hornera

Zadanie sprawdzające opanowanie poprzedniej lekcji

- Dany jest tekst, który należy zaszyfrować. Napisz program, który poda szyfrogram dla podanego tekstu po zastosowaniu szyfru atbash.

Czynności nauczyciela:

- omawia sposób reprezentacji wielkich liczb w tablicy oraz przy pomocy stringów.
- przypomina w jaki sposób można dokonać konwersji wartości cyfry z typu char na int i ze zmiennej typu int na char
- analizuje działania pisemne realizowane na lekcjach matematyki – dodawanie, odejmowanie, mnożenie,
- omawia schemat Hornera – służący do obliczania wartości wielomianu oraz konwersji liczb z systemu dziesiętnego na binarny,

Zadania do rozwiązania na lekcji:

- Napisz program, który po wczytaniu 2 liczb dziesiętnych (o długości powyżej 30 znaków) obliczy i wypisze ich sumę.
- Napisz program, który po wczytaniu 2 liczb dziesiętnych (o długości powyżej 30 znaków) obliczy i wypisze ich różnicę.

Zadania domowe:

- Napisz program, który po wczytaniu 2 liczb dziesiętnych (o długości powyżej 30 znaków) obliczy i wypisze ich iloczyn.
- Napisz program obliczający 11^n , gdzie n podane jest z klawiatury. ($n \leq 100$)

Zadania na sprawdzarce:

- Dodawanie binarne – należy wczytać dwa stringi zbudowane z 0 i 1, a następnie wypisać wartość sumy tych liczb w zapisie binarnym.
- Przechwycona wiadomość – dany jest szyfrogram, który utworzono stosując jeden z szyfrów podstawieniowych. Wiadomo również że jednym ze słów, które zostało zaszyfrowane jest słowo „Nie”. Podaj odszyfrowany tekst.
- Dany jest tekst do zaszyfrowania szyfrem Cezara z przesunięciem k ($0 \leq k \leq 10000$). Podaj szyfrogram wczytanego tekstu jawnego.

Uwagi do realizacji:

Temat nie powinien przysporzyć trudności. Zwykle problemy mają naturę techniczną, nie merytoryczną.

Zajęcia 10 (2 godziny)

Temat: Funkcje w C++, Parametry funkcji oraz zasięg zmiennych. Rekurencja i funkcje rekurencyjne.

Treści z sylabusu:

Matematyka:	Programowanie:	Algorytmika
Systemy liczbowe – reprezentacja liczb w komputerze,	Parametry funkcji oraz zasięg zmiennych, Funkcje rekurencyjne	Algorytm Euklidesa (rekurencyjny), silnia, potęgowanie (liniowe), potęgowanie szybkie, liczby Fibonacciego

Zadanie sprawdzające opanowanie poprzedniej lekcji

- Napisz program obliczający wartość w systemie dziesiętnym liczby zapisanej w systemie liczbowym o podstawie 2 korzystając ze schematu Hornera.

Czynności nauczyciela:

- omawia zasady tworzenia funkcji w C++,
- opisuje typy funkcji, parametry funkcji i zwracane wartości
- wyjaśnia pojęcie rekurencji i sposób zapisu funkcji rekurencyjnych na przykładzie n^k oraz $n!$.
- pokazuje w jaki sposób przy pomocy rekurencji można odwrócić kolejność wypisywanych cyfr podczas konwersji z systemu dziesiętnego na binarny
- opisuje algorytm potęgowania szybkiego P->L oraz L->P

Zadania do rozwiązania na lekcji:

- Napisz program i przeanalizuj czas obliczania rekurencyjnego obliczania n-tej liczby Fibonacciego
- Zaimplementuj algorytm szybkiego potęgowania – obliczenia n^k z wykorzystaniem rekurencyjnego algorytmu szybkiego potęgowania P->L.

Zadania domowe:

- Zaimplementuj algorytm szybkiego potęgowania w wersji L->P z wykorzystaniem schematu Hornera.

Zadania na sprawdzarce:

Liczby zaprzyjaźnione – napisz program, który wypisze liczby zaprzyjaźnione z przedziału $\langle a, b \rangle$
Liczby pierwsze - Masz dany przedział liczb $\langle a, b \rangle$ wypisz wszystkie liczby pierwsze z podanego przedziału.

Dwumian Newtona – masz dane dwie liczby n i k . Oblicz $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Uwagi do realizacji:

Ważny temat, wymagający powtórek i utrwalania. Rekurencja dla młodzieży wydaje się łatwa w zapisie, ale bardzo trudna w analizie działania. Warto wrócić na kolejnych spotkaniach i przeanalizować działanie różnych funkcji rekurencyjnych

Literatura i inne zasoby edukacyjne

Cormen Thomas H., Leiserson Charles E., Rivest Ronald L. – „Wprowadzenie do algorytmów”

Lipski Witold – „Kombinatoryka dla programistów”

Grębosz Jerzy – „Symfonia C++ standard”

Idziaszek T., Diks K, Łącki J., Radoszewski J – „Przygody Bajtazara. 25 lat Olimpiady Informatycznej”

Tomasiewicz Jacek – „Zaprzyjaźnij się z algorytmami”

Serwisy z zadaniami online: <https://codeforces.com>, <https://pl.spoj.com>,
<https://main2.edu.pl>, <https://szkopul.edu.pl/>

Serwisy z wykładami: <https://www.geeksforgeeks.org>, <http://www.algorytm.org/>