Advent of Code:

* Funkcje na stringach (split, join), rozszyfrowywanie (logiczne myślenie, pętle i cięcia stringa, rzutowanie, podstawy)
* Klasa statyczna z f. statyczna
* Dziedziczenie (wirtualna metoda)
* Odczyt z pliku
* Praca na liście
* Praca z Tuples, aby zapisać klawiaturę cyfr 1-9 z uwzględnieniem ich współrzędnych)
* Jakieś FlagAttribute do Stringa => Split(char[], **StringSplitOption.RemoveEmptyEntries**)
* LINQ (Select (o => ..).ToList() umożliwiło bez jawnej pętli zparsować stringi na inty danej listy
* Praca ze słownikiem :

Dictionary<char, int> partOfCheckSum = new Dictionary<char, int>();

partOfCheckSum[letter]++; po key wyszukaj I value++

Lab 5 i 6 to programiki w c++

* Dziedziczenie
* Przeciążanie operatorów
* Praca ze wskaźnikami, na lisach i tablicach
* Biblioteka IO
* Quicksort()

EventAsker

* Czym jest aplikacja RESTful Mówiąc najprościej, w stylu architektonicznym REST dane i funkcjonalność są uważane za zasoby i są dostępne za pomocą identyfikatorów URI (Uniform Resource Identifier). Na zasoby wykorzystuje się zestaw prostych, dobrze zdefiniowanych operacji. Klienci i serwery wymieniają reprezentacje zasobów przy użyciu standardowego interfejsu i protokołu - zwykle HTTP. Zasoby są oddzielone od ich reprezentacji, dzięki czemu ich zawartość jest dostępna w różnych formatach, takich jak HTML, XML, zwykły tekst, PDF, JPEG, JSON i inne. Metadane dotyczące zasobu są dostępne i wykorzystywane na przykład do kontrolowania buforowania, wykrywania błędów transmisji, negocjowania odpowiedniego formatu reprezentacji oraz przeprowadzania uwierzytelniania lub kontroli dostępu. Co najważniejsze, każda interakcja z zasobem jest bezstanowa (każde zapytanie może być interpretowane w oderwaniu od pozostałych).
* JSON

Mój projekt web api z et 6:

* ASP.NET MVC creates the HTML page.
* ASP.NET Web API handles the AJAX requests and returns JSON data.
* Knockout.js data-binds the HTML elements to the JSON data.
* Entity Framework talks to the database.