W poniższych zadaniach skorzystaj z Event Structure, Property Node i Invoke Node, oraz referencji.

Zad. 7.1

a) Napisz program wzorowany na programie "Kalkulator" systemu Windows. Program powinien umożliwiać dokonywanie tych samych operacji (możesz zaniedbać funkcjonalność Memory) poprzez wciskanie odpowiednich przycisków na panelu frontowym, bądź poprzez wciskanie przycisków na klawiaturze. Pamiętaj, by obsłużyć też m.in. klawisze Backspace i Enter.

Na przykład: wciśnięcie kombinacji: 3, 2, Shift+8, 4, Enter – powinno zwrócić wynik równania 32 x 4 czyli 128.

Podpowiedź: Zgrupuj cyfry w cluster i wyłapuj zdarzenie dokonane na którymkolwiek z elementów tego clustra. W celu szybkiego dowiedzenia się, która cyfra została wciśnięta możesz też skorzystać z funkcji *Cluster to array,* szczególnie przy odpowiedniej kolejności kontrolek z clustrze.

b) Jeżeli użytkownik zdecydował się korzystać z klawiatury, podświetlaj na chwilę przyciski, które wybrał (np. gdy użytkownik wciśnie „1" na klawiaturze, mrugnij przyciskiem „1" na front panelu), tak jak to robi kalkulator Windows.

c) Zbuduj subVI, który wywołany w momencie inicjalizacji programu pozwoli na ustawienie koloru tła wszystkich przycisków z cyframi znajdującymi się w VIu głównym kalkulatora, oraz osobnego subVI do ustawiania tła wszystkich przycisków funkcyjnych (takich jak np. =, %, +). Kolor powinien być jednym z argumentów subVI.

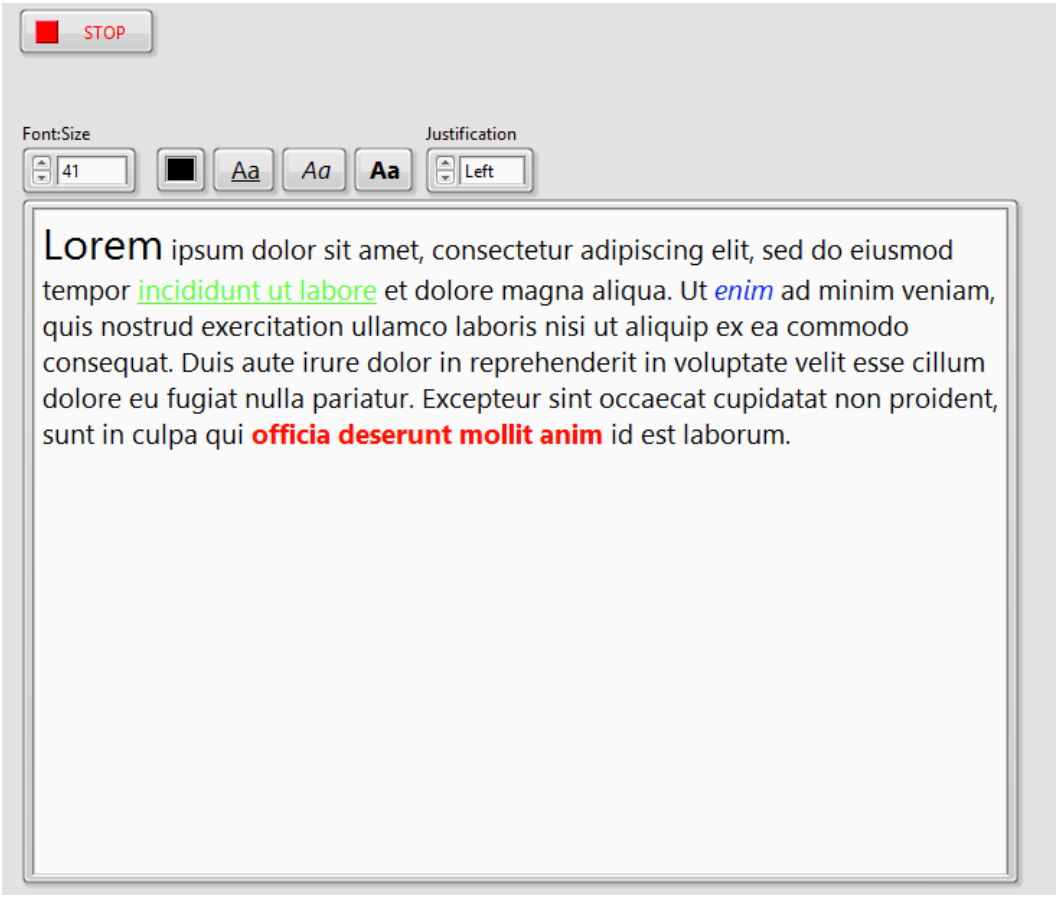
Podpowiedź: Zapoznaj się z różnicą między Implicit, a Explicit Property Node oraz zapoznaj się z referencjami do kontrolek.

Przydatne linki:
Property Node definition
Explicit vs Implicite Property Node LabVIEW
How to use Property Node in subVI

Zad. 7.2

Napisz program – edytor tekstu, pozwalający użytkownikowi na wprowadzanie/usuwanie tekstu, a także na zaznaczenie części tekstu i jego: pogrubienie/przechylenie, zmianę wielkości czcionki, zmianę koloru czy wyrównanie tekstu, z wykorzystaniem odpowiednich przycisków. Po zaznaczeniu danej części tekstu, zmiana wielkości czcionki powinna być też możliwa z wykorzystaniem scrolla myszki.

Użytkownik powinien też móc zaznaczyć cały tekst z wykorzystaniem skrótu klawiszowego ctrl+a, skopiować tekst z wykorzystaniem ctrl+c, oraz go wkleić z wykorzystaniem ctrl+v. Wciskając ctrl+z powinien też móc cofnąć maksymalnie 10 ostatnio wprowadzonych znaków.



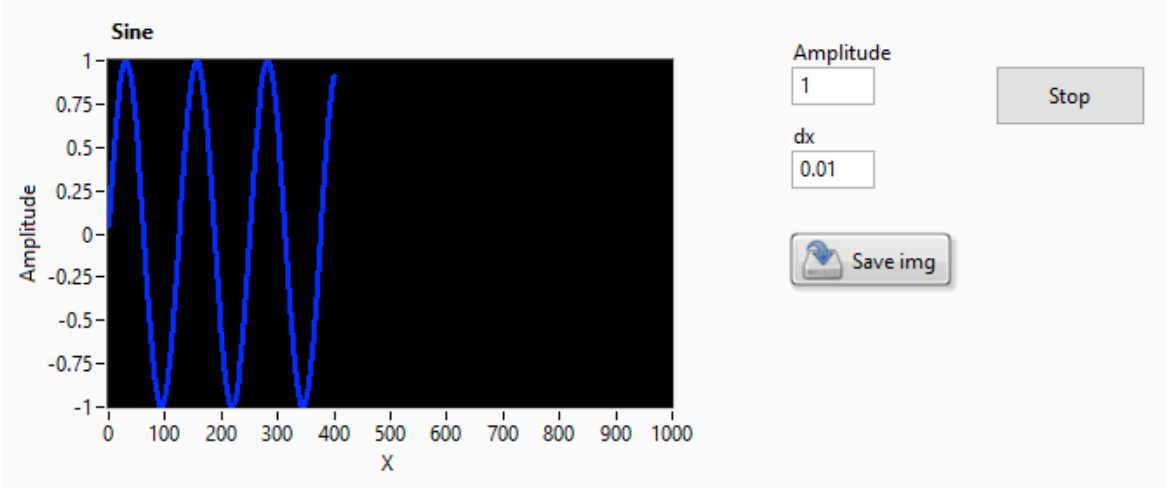
Zad. 7.3

Napisz program, pozwalający użytkownikowi na generowanie przebiegu cosinusa. Cosinus powinien być generowany na bieżąco – w trakcie działania programu – i dostosowywać się jeżeli jego parametry (amplituda, dx) zostaną w trakcie działania programu zmienione. Na wykresie wyświetlaj ostatnich 1k próbek przebiegu.

a) Na wciśnięcie przycisku „Save img" aktualnie wyświetlany przebieg powinien zostać zapisany w folderze *Images* tworzonym na pulpicie w momencie uruchamiania programu, przy założeniu, że folder jeszcze nie istnieje. W momencie zapisu program powinien sugerować nazwę pliku: Image_000.bmp (bądź o odpowiednio większym indeksie, jeżeli w folderze znajdują się już jakieś pliki) oraz sugerować lokalizację zapisu – folder *Images*. Do zrzucenia ekranu wykresu wykorzystaj odpowiedni Invoke Node.

b) W momencie uruchomienia programu, wartości wszystkich kontrolek na front panelu powinny zostać zresetowane do ich ustawień domyślnych. W tym podpunkcie NIE inicjalizuj kontrolek, używając dla każdej z nich Property Node Value z osobna, ale zainicjalizuj wartości kontrolek w VIu wykorzystując jego Invoke Node.

Wygląd panelu frontowego programu:



Zawartość pliku po otwarciu przez domyślny program Windowsa: