Kalkulator dla dużych liczb i systemów liczbowych o podstawie 2-16

autor: Marek Mytkowski

---------------------------------------------------------------------------------------------------------------------------

**Uruchomienie programu**

Uruchomienie programu następuje poprzez uruchomienie go razem z argumentami lub bez nich (np. w przypadku windows - w cmd)

Ścieżki do plików należy podawać względem pliku wykonywalnego lub bezwzględnie (pełne ścieżki)

Argumenty uruchomienia kalk.exe:

Brak argumentow:

domyślnie ustawia ścieżkę do pliku wejściowego na "../Input/input.txt" znajdującego się w folderze Input

oraz ścieżkę pliku wyjściowego na "output.txt" znajdującego się w folderze programu (Exe)

-n

ustawia niestandardowe nazwy pliku (wejsciowego i wyjsciowego)(jeśli nie zostanie podana ścieżka do pliku to plik jest

traktowany jakby był w folderze w którym znajduje się program) według wzoru:

"kalk.exe -n <nazwa (opcjonalnie ze ścieżką do) pliku wejściowego> <nazwa (opcjonalnie ze sciezka do) pliku wyjściowego>"

-p

dzieli określony plik wejściowy na oddzielne operacje (każda w innym pliku, nazwane liczbami porządkowymi operacji). Wszystkie pliki automatycznie tworzone są w folderze gdzie znajduje się plik wykonywalny exe i nie ma możliwości zmiany tego folderu

"kalk.exe -p <nazwa (opcjonalnie ze ścieżką do) pliku wejściowego>

--help

wyświetla zawartość pliku readme.txt

UWAGA: kalk.exe z argumentem -p nie usuwa poprzednio utworzonych przez niego plików, należy zrobić to ręcznie

UWAGA 2: nazwy plików prosze podawac z rozszerzeniem

---------------------------------------------------------------------------------------------------------------------------

**Obsługa błędnych danych**

W przypadkach opisanych poniżej program obsługuje niepoprawne dane:

-gdy nie znaleziono pliku podanego jako argumentu ( lub w przypadku wywołania bez argumentów nie znaleziono input.txt) lub nie rozpoznano opcji uruchomienia, program drukuje komunikat i kończy się wykonywać z kodem 0 (todo)

-gdy wprowadzono nieznana operacje to program drukuje komunikat i kończy się wykonywać z kodem 0

-gdy wprowadzono zbyt duza podstawę systemu liczbowego program drukuje komunikat i kończy się wykonywać z kodem 0

-gdy wprowadzono niepoprawna liczbę program wyświetla komunikat i kończy się wykonywać z kodem 0

-gdy występuje dzielenie przez zero lub modulo przez zero program drukuje komunikat i kończy się wykonywać z kodem 0

---------------------------------------------------------------------------------------------------------------------------

**Opis danych wejsciowych i wyjsciowych:**

Dane wejściowe:

Wzór dla operacji arytmetycznych:

---------------------------------------------------------------------------------------------------------------------------

<symbol> <podstawa systemu liczbowego liczb>

<argument 1>

<argument 2>

---------------------------------------------------------------------------------------------------------------------------

Operacje arytmetyczne możliwe do wykonania:

Dodawanie - symbol: "+"

Dzielenie całkowite - symbol: "/"

Reszta z dzielenia (modulo) - symbol: "%"

Mnożenie - symbol: "\*"

Potęgowanie - symbol: "^"

Wzór dla konwersji między systemami liczbowymi:

---------------------------------------------------------------------------------------------------------------------------

<podstawa systemu liczbowego liczby konwertowanej> <docelowa podstawa systemu liczbowego>

<konwertowana liczba>

---------------------------------------------------------------------------------------------------------------------------

Kolejne operacje należy podać w jednym pliku pod sobą

Dane Wyjściowe:

W zależności od argumentów uruchomienia w jednym pliku pod sobą lub oddzielnie każda operacja w innym pliku według schematu:

Dla każdej operacji arytmetycznych:

---------------------------------------------------------------------------------------------------------------------------

<symbol> <podstawa systemu liczbowego liczb>

<argument1>

<argument2>

<wynik>

---------------------------------------------------------------------------------------------------------------------------

Dla każdej konwersji między systemami liczbowymi:

---------------------------------------------------------------------------------------------------------------------------

<podstawa systemu liczbowego liczby konwertowanej> <docelowa podstawa systemu liczbowego>

<konwertowana liczba>

<przekonwertowana liczba>

---------------------------------------------------------------------------------------------------------------------------

**Kompilacja**

Plik wykonywalny programu (kalk.exe) został skompilowany w wersji release x64 przy pomocy programu Microsoft Visual Studio 2022

Current na podstawie pliku źródłowego main.c w systemie Windows 10 64-bit

---------------------------------------------------------------------------------------------------------------------------

**Opis struktury programu**

Program by wykonywać operacje na bardzo dużych liczbach przetrzymuje je w pamięci jako dynamicznie alokowane ciągi znaków i pracuje na nich właśnie w takiej formie. Stworzone tablice globalne i ich uzupełnianie na początku wykonywania programu służą temu by zamieniać char na liczby w systemie 10 i na odwrót (np. n\_to\_c[0] = ‘0’ albo c\_to\_n[(int) ‘0’] = 0) Wczytywanie następuje po jednym znaku z pliku wejściowego szukając znaków charakterystycznych jak znak końca linii albo końca pliku. Program rozdziela w swojej strukturze sytuację operacji algebraicznych i konwersji ze względu na to, że bardzo się różnią pod względem danych i ich zapisu. Operacje arytmetyczne są wykonywane w taki sposób jakby były liczone pisemnie. Funkcje odpowiadające za wykonywanie operacji arytmetycznych lub konwersji zwracają dynamicznie zaalokowany ciąg znaków będący wynikiem lub przekonwertowaną liczbą, jednocześnie nie zwalniają pamięci argumentów - jest to wykonywane na zewnątrz funkcji.