Sprawozdanie/ Dokumentacja

Wykonywał:	Maciej Sołtys / Aleksander Taraska
Data wykonania:	04.12.2025
Czas wykonania:	Kilka tygodni
Losowa liczba:	479823

KALKULATOR U.M.S.C.

ZARYS DOKUMENTU - ZMIENIĆ

1. Funkcje Użytkowe

NAZWA FUNKCJI	ARGUMENTS	RETURN
clearInputTextAndSlice	std::string	std::vector<char>
encode	std::vector<char>	std::vector<int>
decode	std::vector<int>	std::vector<char>
changeToBase10Number	int, std::vector<int>	unsigned long long
changeBase	int, std::vector<int>, int	std::vector<int>
add	int, std::vector<int>, std::vector<int>, std::vector<int>*	std::vector<int>

2. Pozostałe funkcje

NAZWA FUNKCJI	ARGUMENTS	RETURN
fastPow	unsigned long long, unsigned long long	unsigned long long
bigLetters	char	char
isIllegalSymbol	int, std::vector<int>	bool

SZCZEGÓŁOWY OPIS WYBRANYCH FUNKCJI

Funkcja encode(x)

Funkcja służy do zamiany wektora znaków na wektor reprezentacji liczbowej.

- Tworzony jest nowy wektor liczb.
- Dla każdego elementu wektora wejściowego x wykonywana jest iteracja po tablicy dictionary[].
- W przypadku zgodności znaków, do wektora liczbowego dodawany jest indeks odpowiadający pozycji znaku w dictionary[].
- Po zakończeniu iteracji zwracany jest wektor wartości liczbowych odpowiadających znakom wejściowym.

Funkcja decode(x)

Funkcja odwrotna do encode(x), zamieniająca wektor reprezentacji liczbowej na wektor znaków.

- - Przyjmuje wektor liczb jako argument.
- - Sprawdza, czy każda wartość mieści się w zakresie aktualnej podstawy.
- - Jeśli warunek jest spełniony, pobierany jest odpowiedni znak z tablicy dictionary[].
- - Tworzony jest nowy wektor znaków, który po zakończeniu iteracji zostaje zwrócony jako wynik.

Funkcja changeSystem(base, x, wantedBase)

Funkcja realizuje konwersję liczby pomiędzy różnymi systemami liczbowymi.

- - Sprawdzana jest poprawność podstawy docelowej wantedBase (czy > 1 oraz ≤ maxBase).
- - Walidowany jest wektor x – wszystkie cyfry muszą mieścić się w zakresie podstawy base.
- - Wektor x zostaje przeliczony na wartość w systemie dziesiętnym.
- - Wynik w systemie dziesiętnym jest konwertowany na wektor cyfr w systemie wantedBase.
- - Wektor wynikowy zostaje odwrócony w celu zachowania poprawnej kolejności cyfr.
- - Funkcja zwraca liczbę w nowej podstawie.

Funkcja add(base, x1, x2, carry)

Funkcja realizuje dodawanie dwóch liczb zapisanych w tej samej podstawie, w sposób analogiczny do dodawania pisemnego.

1 Walidacja:

1.1 Sprawdzenie, czy base mieści się w zakresie maxBase.
1.2 Weryfikacja, czy wektory x1 i x2 zawierają poprawne cyfry w danej podstawie.
1.3 Przygotowanie:
1.4 Określenie, który wektor jest dłuższy (top), a który krótszy (bottom).
1.5 W przypadku równej długości: x1 → top, x2 → bottom.
1.6 Do krótszego wektora dodawane są zera z lewej strony w celu wyrównania długości.
1.7 Tworzony jest wektor carry, przechowujący przeniesienia.

2 Dodawanie kolumnowe:

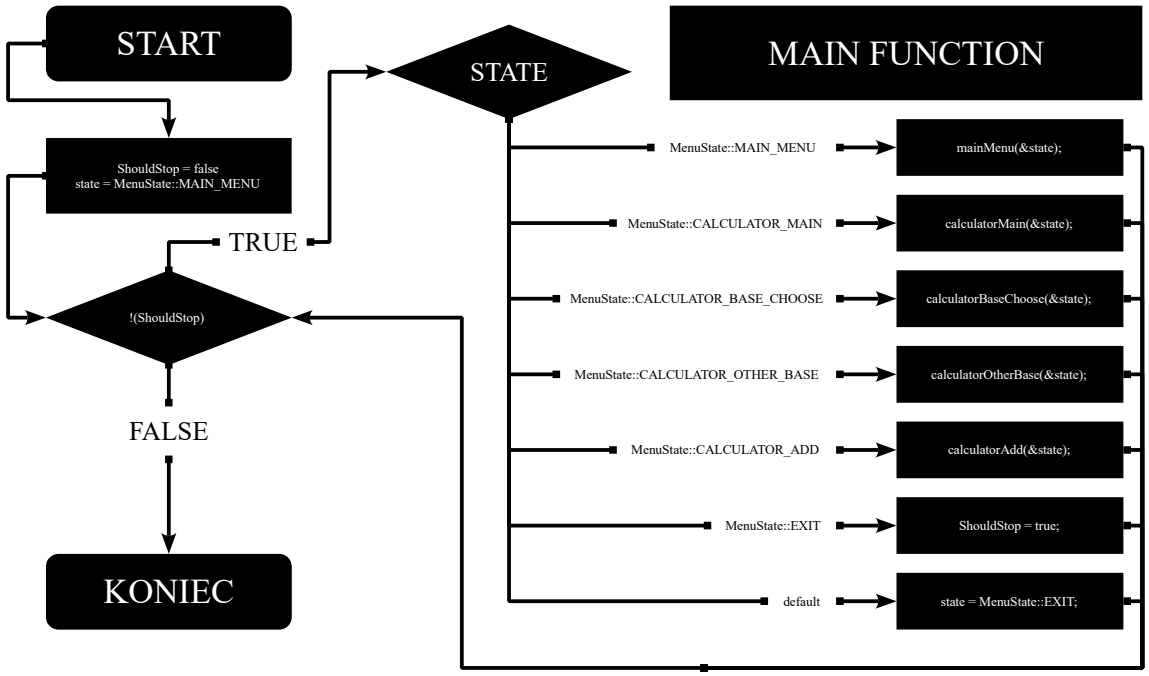
2.1 Iteracja od prawej strony (najmniej znaczącej cyfry).
2.2 W każdej kolumnie obliczana jest suma:
suma=top[i]+bottom[i]+carry[i]- Jeśli suma ≥ base:
2.3 Do wyniku wpisywane jest suma % base.
2.4 Do carry[i+1] zapisywane jest suma / base.
2.5 Jeśli suma < base:
2.6 Do wyniku wpisywana jest wartość suma.
2.7 Zabezpieczenie: przy pierwszej iteracji algorytm nie odwołuje się do carry[-1].

3 Finalizacja:

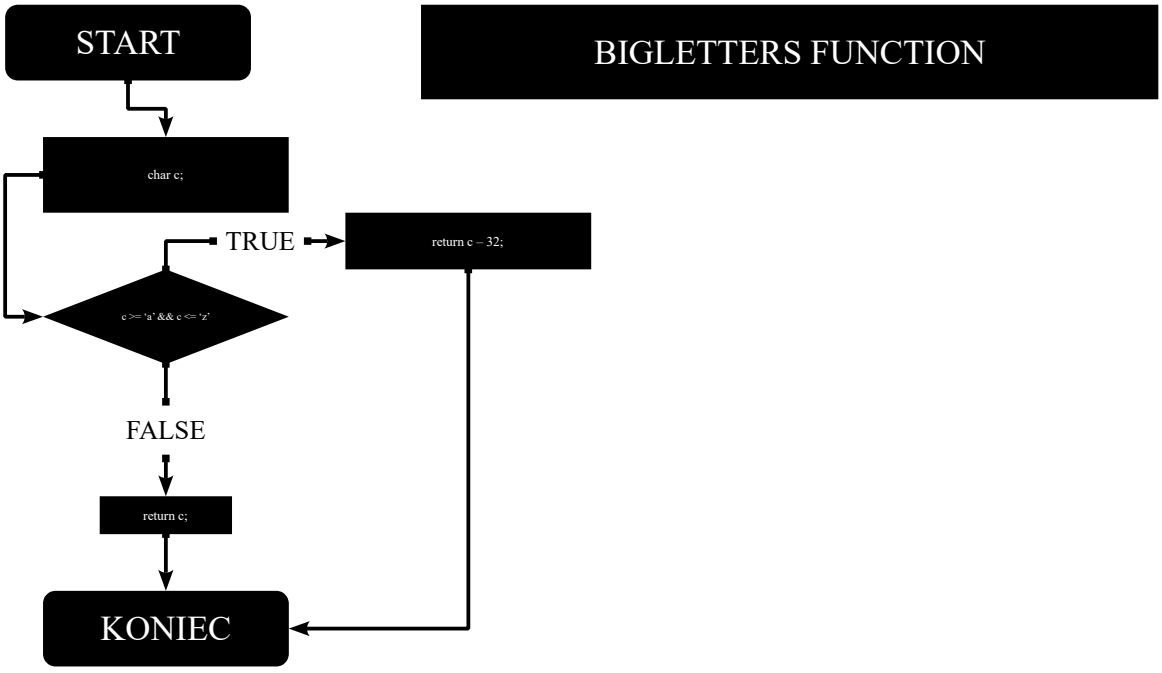
3.1 Jeśli po ostatniej kolumnie pozostaje przeniesienie, dopisywane jest jako dodatkowa cyfra.
3.2 Wektor wynikowy zostaje odwrócony w celu zachowania poprawnej kolejności.
3.3 Funkcja zwraca wektor cyfr będący sumą dwóch liczb.

SCHEMATY BLOKOWE WYBRANYCH FUNKCJI

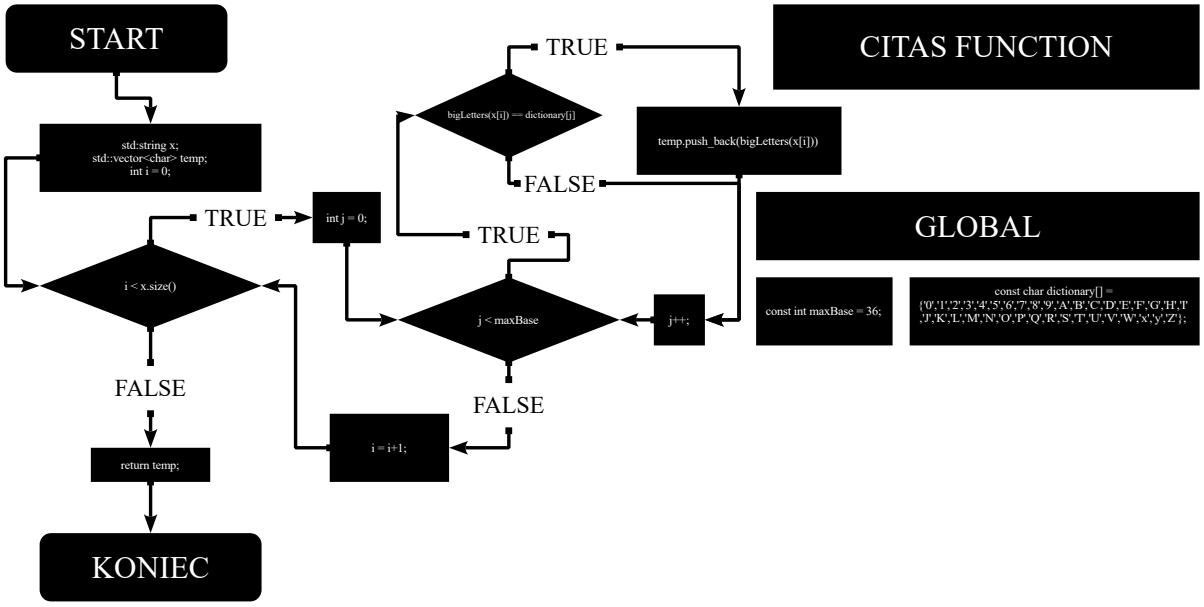
1. Schemat blokowy funkcji Main, z funkcją wyboru karty menu.



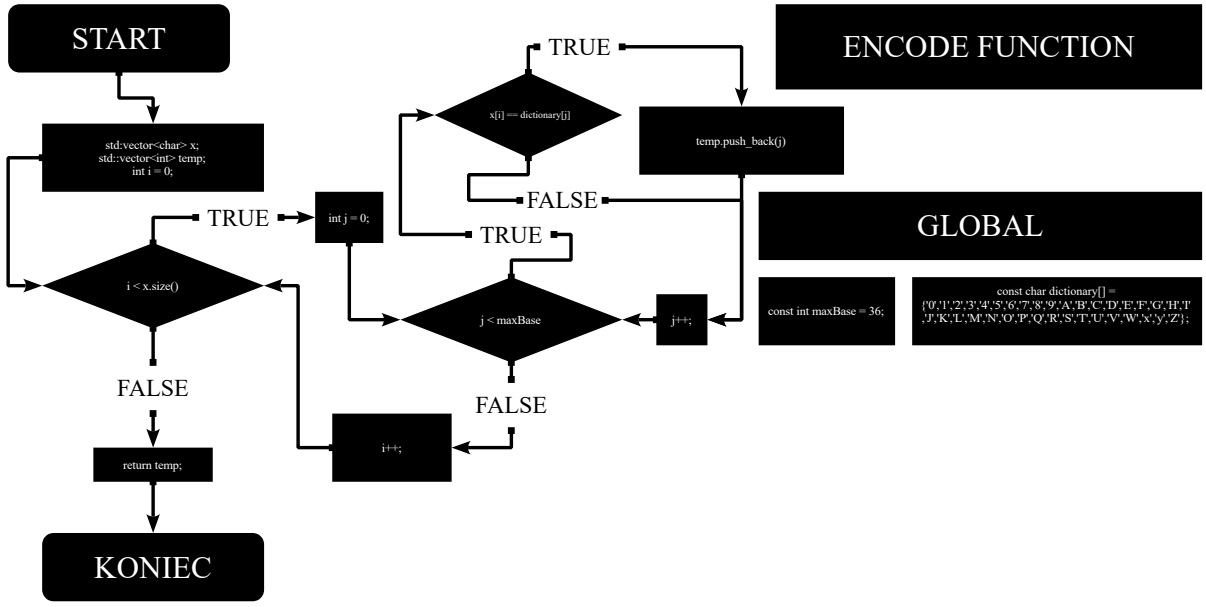
2. Schemat blokowy funkcji bigLetters, która zamienia małe litery na duże.



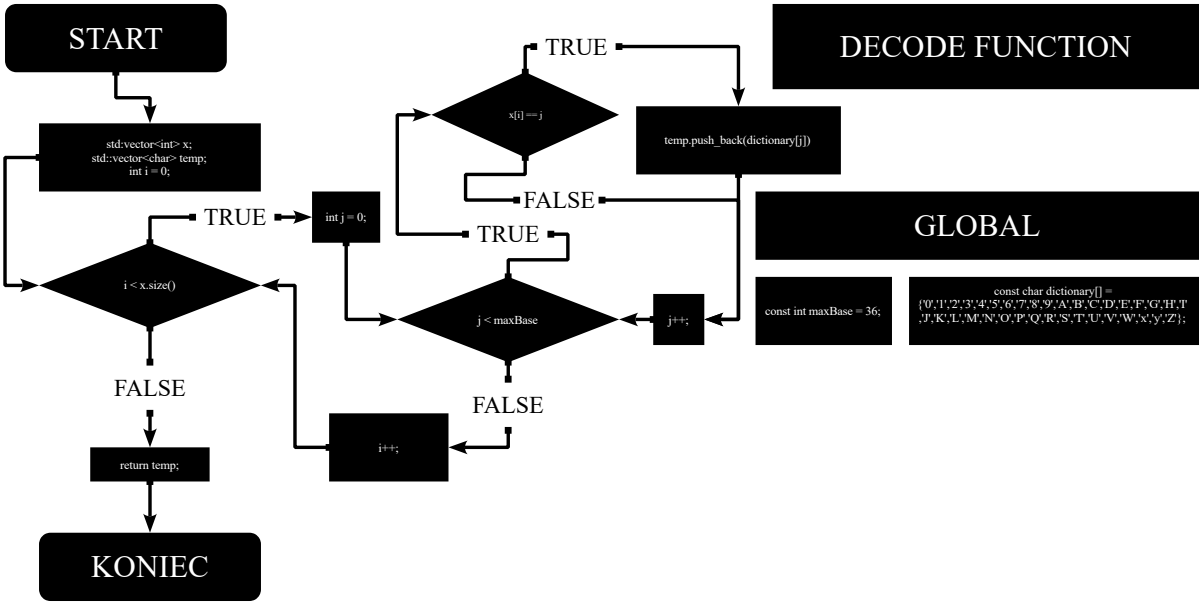
3. Schemat blokowy funkcji C.I.T.A.S. clearInputTextAndSlice. Czyści wejście.



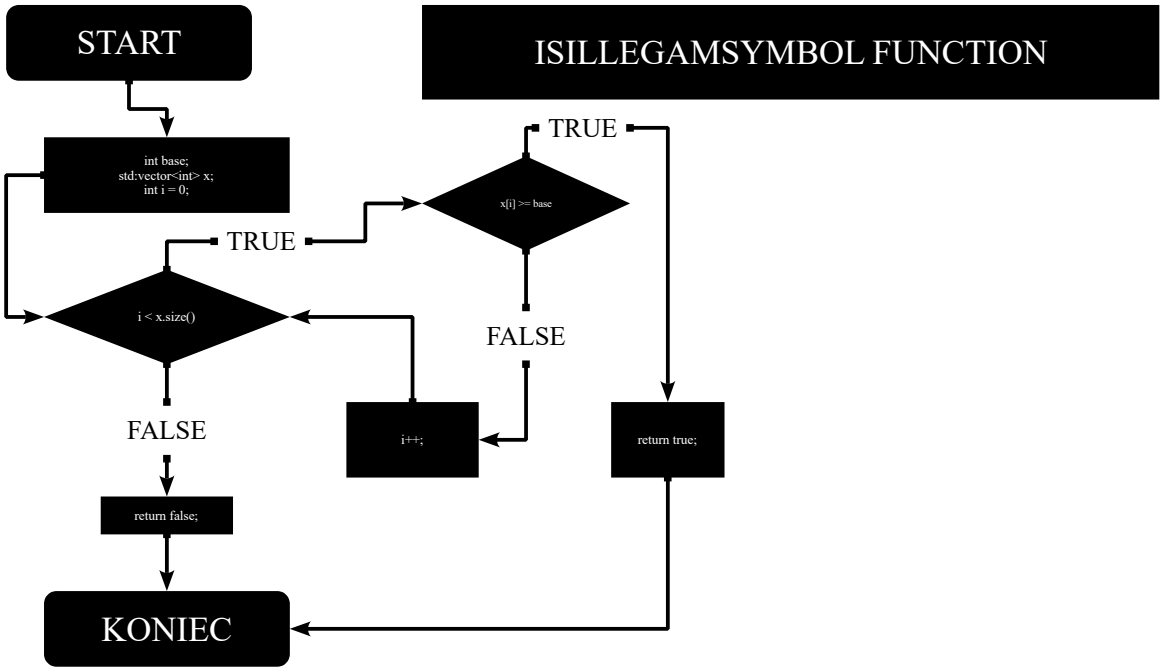
4. Schemat funkcji encode, zamienia znaki na reprezentacje liczbowe.



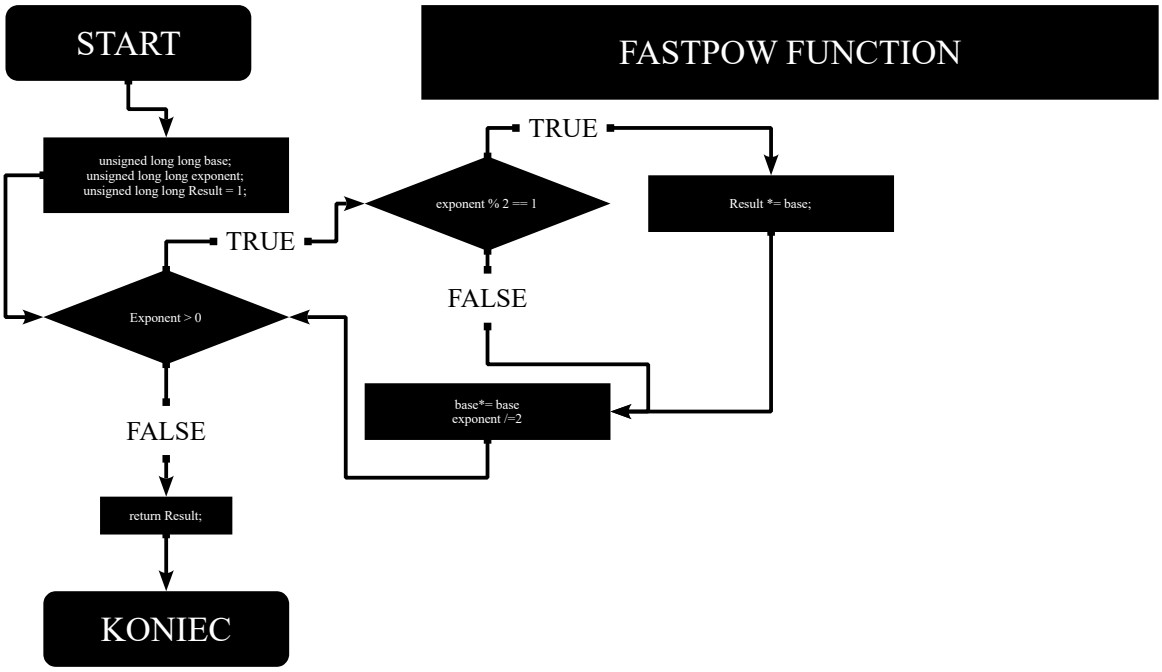
5. Schemat funkcji decode, zamienia reprezentację liczbową na znaki.



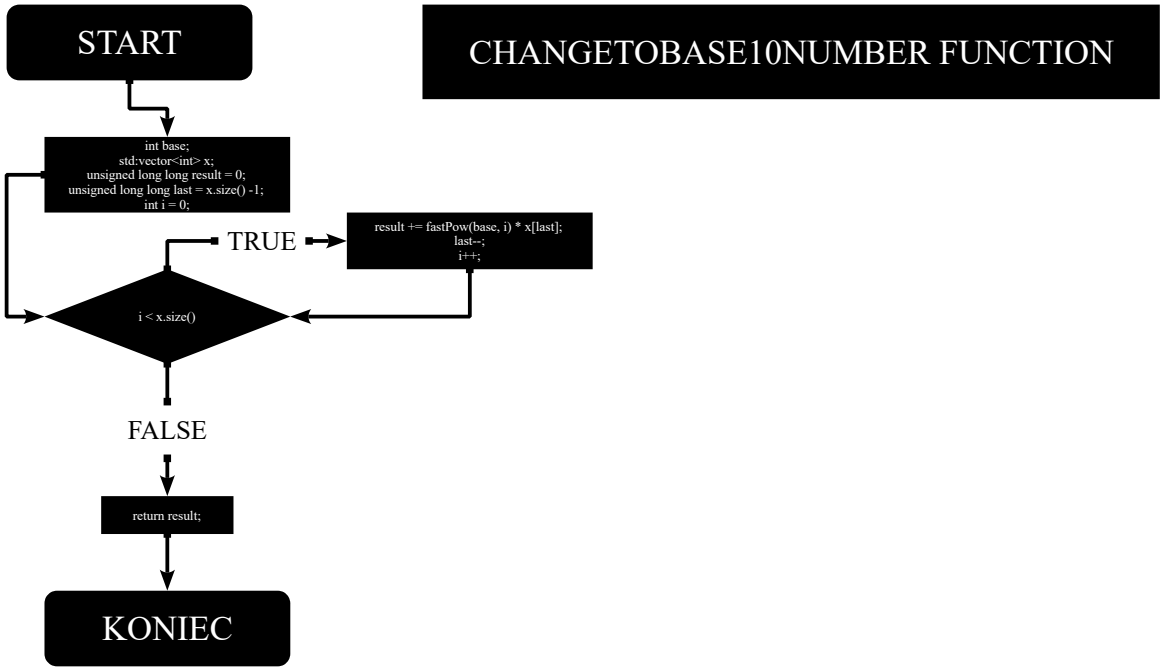
6. Schemat funkcji isIllegalSymbol, sprawdza czy wartości nie są większe niż podstawa.



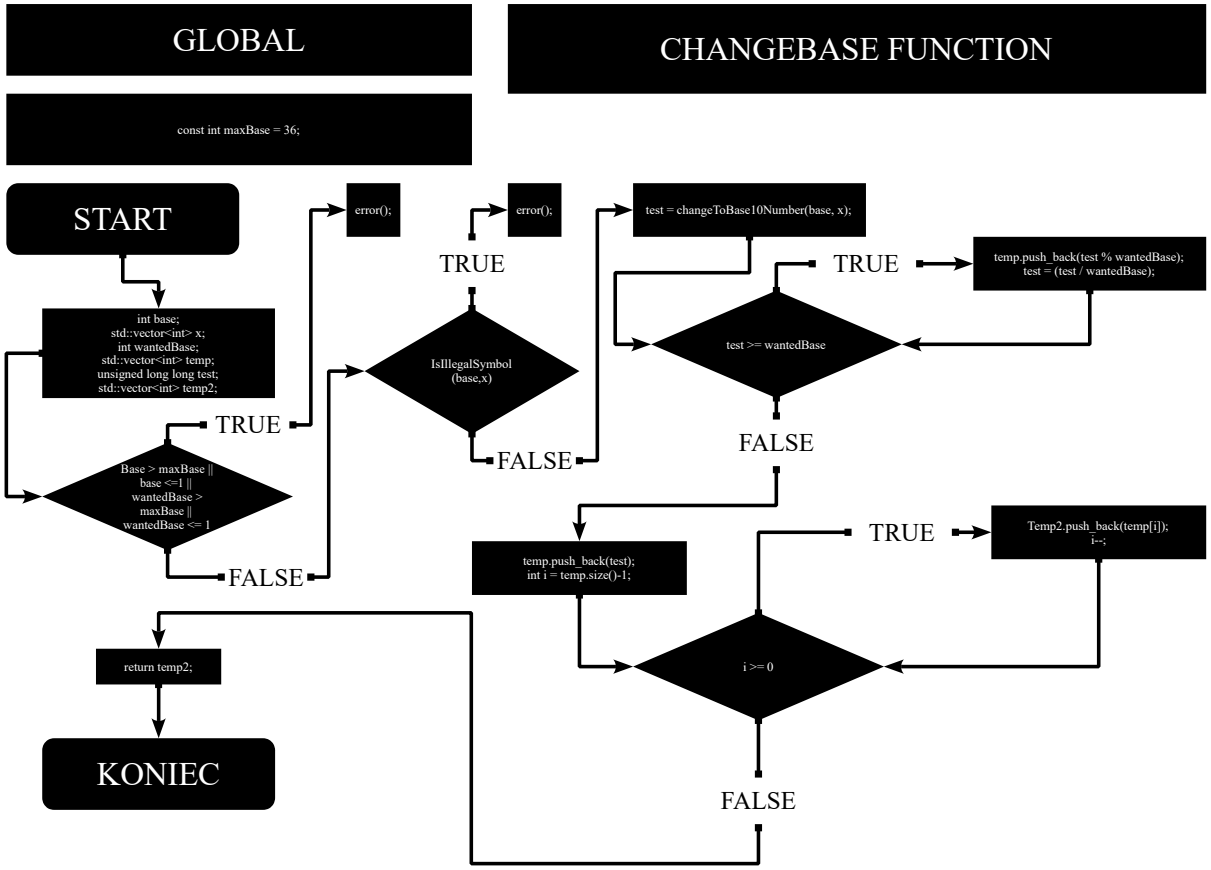
7. Schemat funkcji fastPow, służy do szybkiego potęgowania dużych liczb.



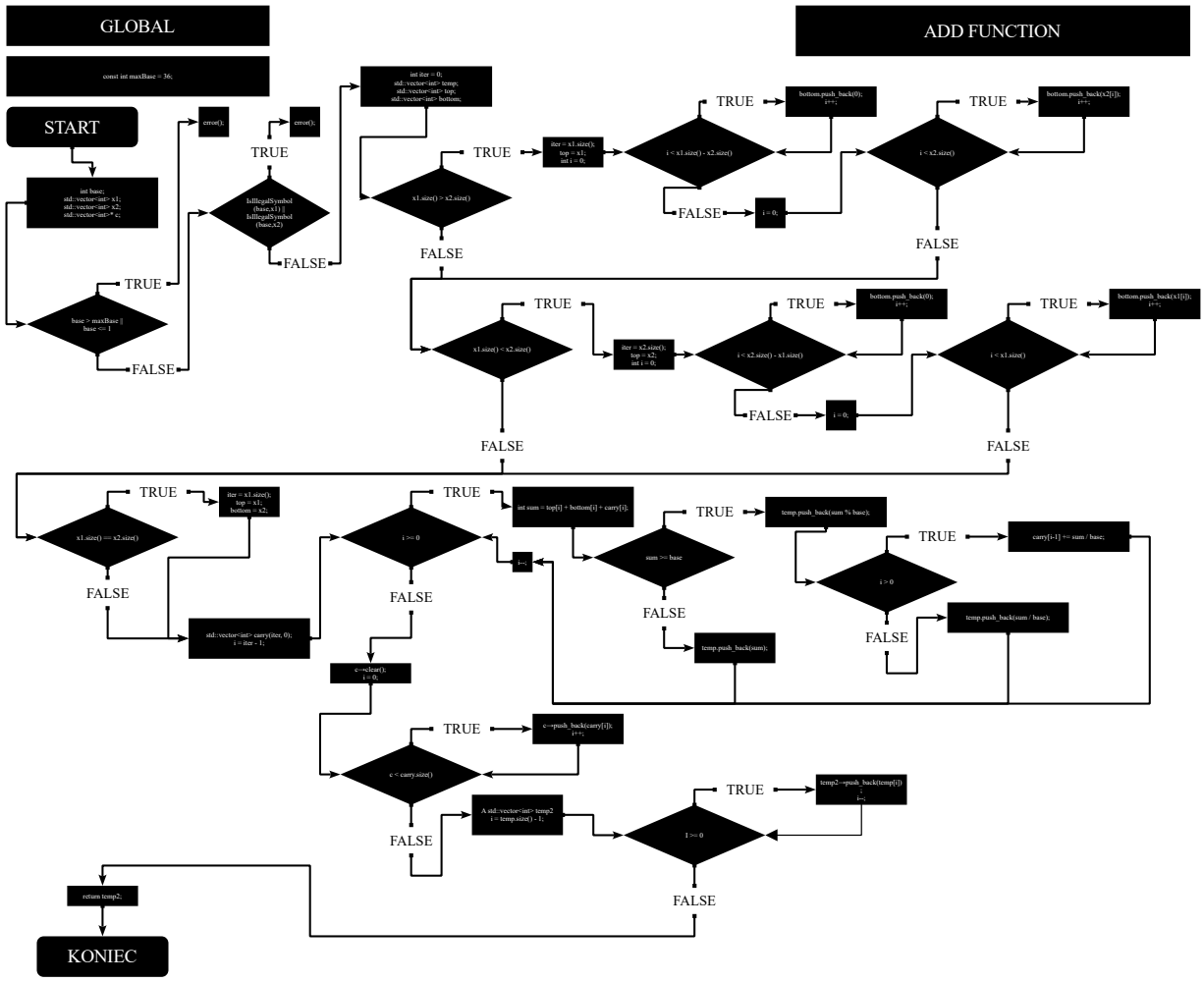
8. Schemat funkcji changeToBase10Number, zamienia liczbę w jakimkolwiek systemie na liczbę w systemie o podstawie 10, wynikiem jest liczba całkowita.



9. Schemat funkcji changeBase, służy do zamiany liczby z jednej podstawy na drugą.

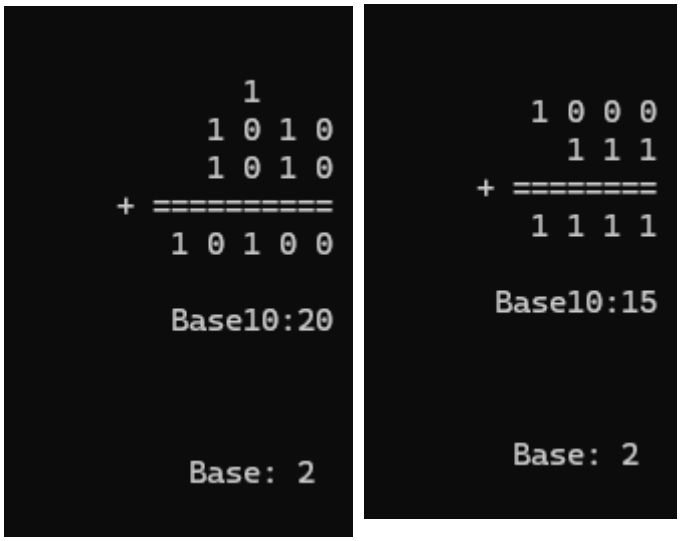


10. Schemat funkcji add, służy do dodania dwóch liczb ze sobą, o tych samych podstawach. Maksymalna liczba która może być reprezentowana to FFFFFFFFFFFFFFF$_{(hex)}$



OBLICZENIA WŁASNE I PROGRAMU

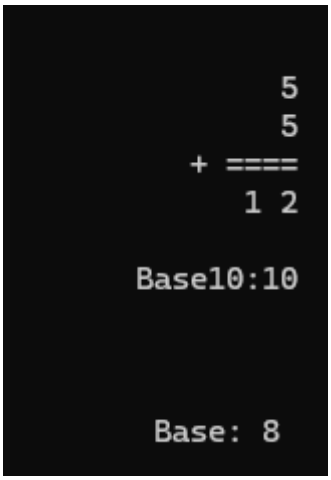
1. System dwójkowy:

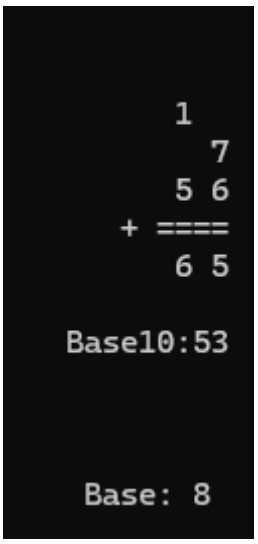


Obliczenia własne:
$1010_{(2)}+1010_{(2)} = 10100_{(2)} = 20_{(10)}$

$1000_{(2)} + 111_{(2)} = 1111_{(2)} = 15(10)$

2. System ósemkowy

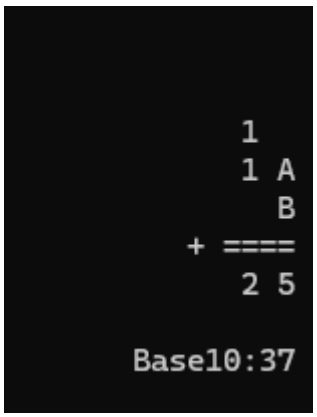


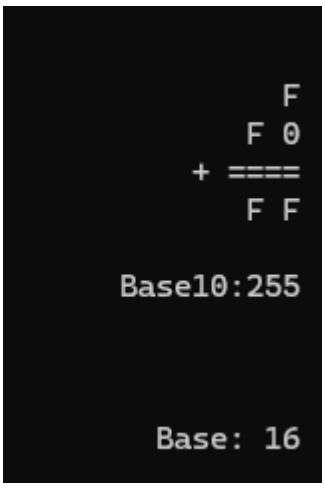


Obliczenia własne:
$5_{(8)}+5_{(8)}=12_{(8)}=10_{(10)}$

$7_{(8)}+56_{(8)}=65_{(8)}=53_{(10)}$

3. System szesnastkowy





Obliczenia własne:
$1A_{(hex)}+B_{(hex)}=25_{(hex)}=37_{(10)}$

$F_{(hex)}+F0_{(hex)}=FF_{(hex)}=255_{(10)}$