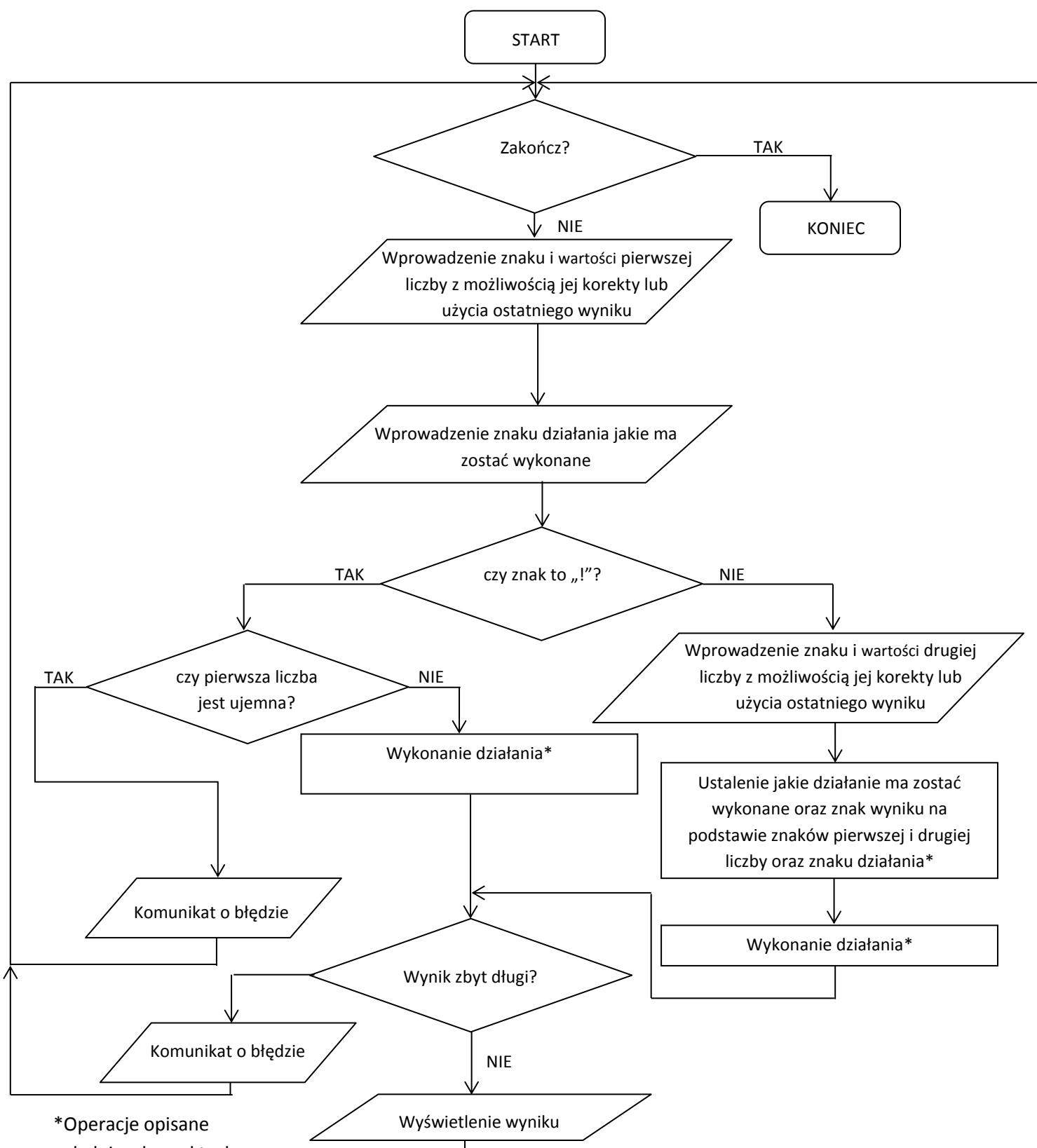


Projekt 2.

Temat projektu:

Wykonać precyzyjny kalkulator, który może operować na liczbach do 70 cyfr. Działania ma realizować to: mnożenie, dodawanie, odejmowanie, silnia. Ma mieć możliwość korzystania z ostatniego wyniku. Wymagana sygnalizacja przekroczenia zakresu oraz komunikaty o błędach.

Schemat blokowy programu:



Zmienne globalne i makrodefinicje :

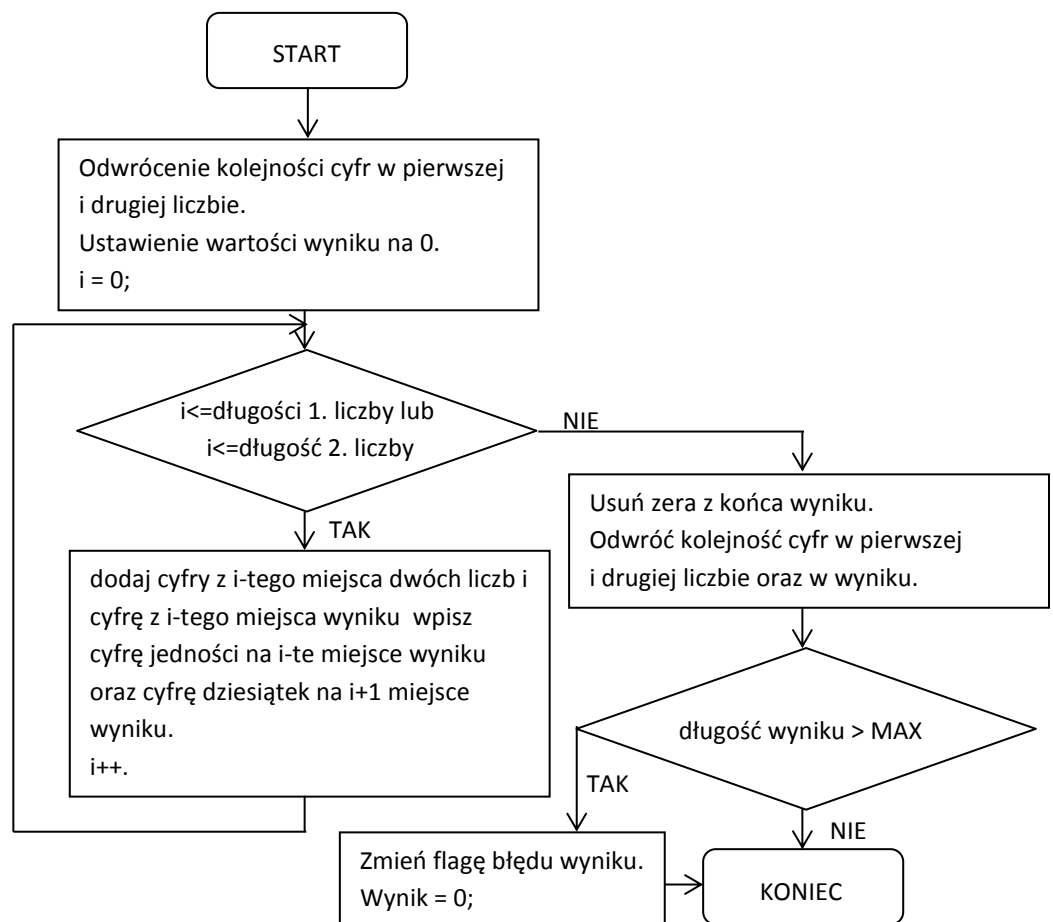
#define MAX 70	zakres długości liczb.
struct number{ char value[MAX+2]; char sign; char end; }	struktura – liczba tablica znakowa przechowująca wartość liczby zmienna znakowa przechowująca znak liczby (+ lub -) zmienna pomocnicza przy wczytywaniu liczby, w przypadku wyniku flaga zbyt dużej ilości cyfr;
number result;	zmienna przechowująca wynik, na niej wykonywane są operacje przez funkcje wykonywania działań

Opis wybranych operacji ze schematu blokowego programu:

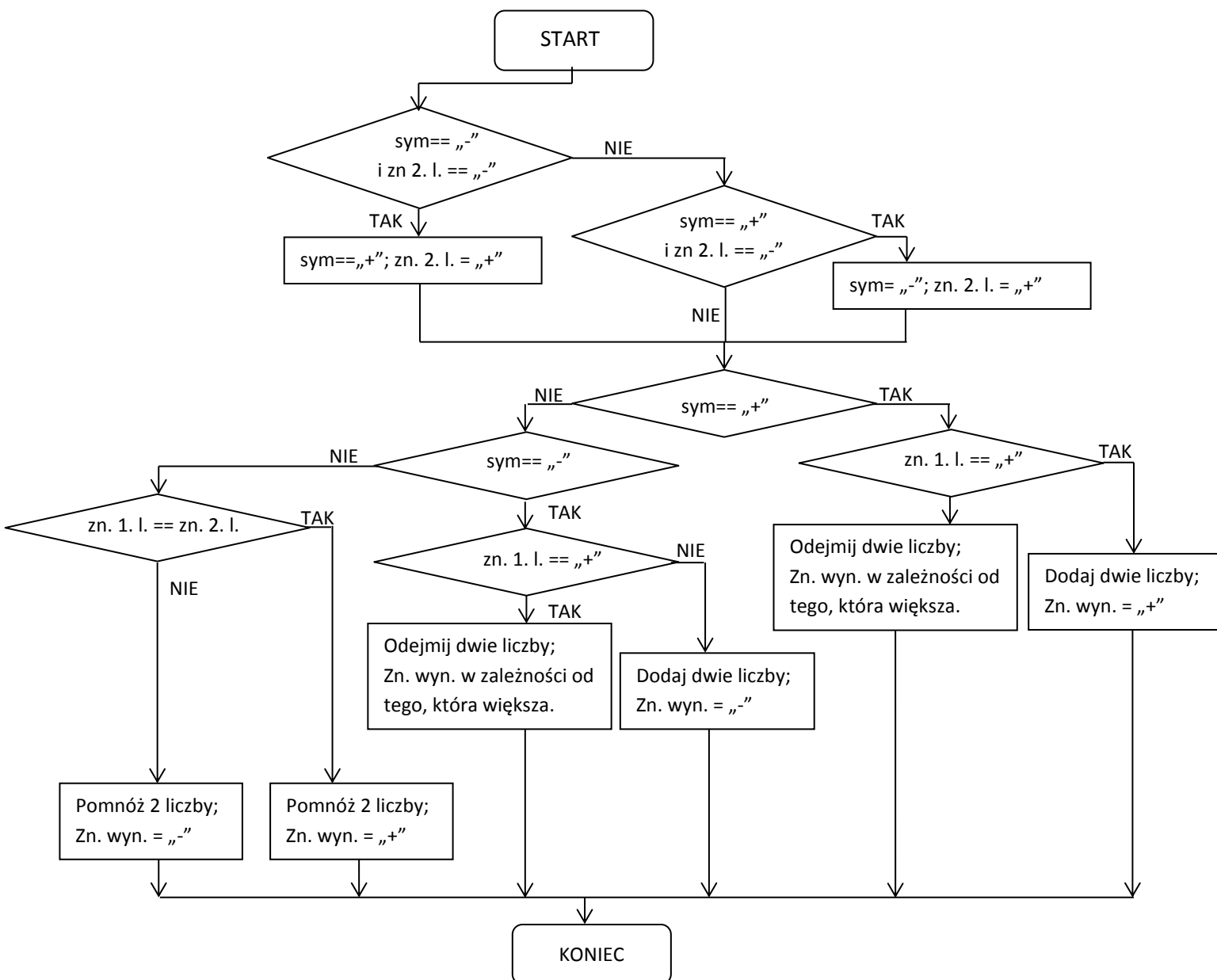
- Wykonanie działania odbywa się za pomocą 4 funkcji:
 - void add(char fnumb[], char snumb[]) – dodawanie dwóch liczb
 - void sub(char fnumb[], char snumb[]) – odejmowanie pierwszej liczby od drugiej (pierwsza liczba musi być większa od drugiej)
 - void mul(char fnumb[], char snumb[]) – mnożenie dwóch liczb
 - void fac(char fnumb[]) - silnia liczby

Wszystkie powyższe funkcje zbudowane są analogicznie. Jako parametry przyjmują tablice znakowe (jedną lub dwie) wypełnione cyframi od 1 do 9. W przypadku funkcji sub pierwsza liczba musi być większa od drugiej. Sprawdzenie poprawności danych odbywa się na poziomie implementacji funkcji. Funkcje powodują zmianę globalnej zmiennej – char result.value[MAX+2], która wyświetlana jest jako wynik działania w końcowej fazie działania programu.

Schemat blokowy funkcji: void add(char fnumb[], char snumb[]):



2. Ustalenie jakie działanie ma zostać wykonane oraz ustalenie znaku wyniku dokonuje się przy pomocy instrukcji warunkowych `case` i `if`. Schemat blokowy tego fragmentu kodu:
 sym – znak działania; zn. 1. l. – znak 1. liczby; zn. wyn. – znak wyniku.



Pozostałe funkcje użyte w programie:

void reverse(char fnumb[])	jako parametr przyjmuje tablicę znakową dowolnej długości i odwraca kolejność znaków w tej tablicy.
void copy(char fnumb[], char snumb[])	jako parametr przyjmuje dwie tablice znakowe dowolnej długości i kopiuje zawartość pierwszej tablicy do drugiej.
void clearTab(char fnumb[], int size)	jako parametr przyjmuje tablicę znakową oraz jej rozmiar i na każdym miejscu jako wartość wpisuje 0.
void clearTab(number fnumb, number snumb, char symb)	jako parametry przyjmuje dwie struktury number oraz znak i wypisuje na konsolę zgodnie z algorytmem: znak pierwszej liczby, pierwsza liczba, znak, znak drugiej liczby, druga liczba. Konieczna z uwagi na błędy związane z wypisywaniem znaku "0" w bibliotece ncurses.

Biblioteki użyte w programie:

Oprócz standardowych bibliotek (<stdio.h>, <stdlib.h>) w programie użyto bibliotek: <string.h>, w celu użycia funkcji: strlen(), strcmp()) oraz biblioteki <ncurses.h>, w celu użycia trybu wprowadzania „bez enteru” oraz podświetlania błędów na czerwono i wyniku na zielono.

