**1. Opis problemu**

Celem projektu było stworzenie kalkulatora podającego wynik danego wyrażenia arytmetycznego, który obsługuje działania dodawania, odejmowania, mnożenia, dzielenia oraz potęgowania z uwzględnieniem kolejności wykonywania działań (włącznie z obsługą nawiasów), działającego również dla liczb niecałkowitych. Program przyjmuje wyrażenie arytmetyczne i zwraca je zapisane w formie ONP oraz podaje jego wynik.

Dla danych:

5.4\*((3-7)^2/3\*-(5+1))-3

Program wypisuje:

zapis w onp: 5.4 3 7 - 2 ^ 3 / 5 1 + ~ \* \* 3 -

wynik: -175.800000

Oraz zapisuje w pliku *historia\_sesji.txt*:

wyrazenie arytmetyczne: 5.4\*((3-7)^2/3\*-(5+1))-3

zapis w onp: 5.4 3 7 - 2 ^ 3 / 5 1 + ~ \* \* 3 -

wynik: -175.800000

Plik *historia\_sesji.txt* przechowuje historię wykonanych w trakcie ostatniego uruchomienia programu wyrażeń algebraicznych. Z każdym uruchomieniem programu usuwana jest historia poprzedniej sesji.

**2. Instrukcja obsługi**

1. Wejście do folderu z programem

1. Kompilacja pliku

Dołącz *-lm* na końcu polecenia *gcc*.

1. Uruchomienie pliku wykonywalnego
2. Przykładowe użycie programu



**3. Specyfikacja wejścia**

Dane przyjmowane są w sposób standardowy. Program obsługuje liczby całkowite oraz niecałkowite zapisane z kropką oznaczającą początek części ułamkowej. Liczby oraz operatory nie mogą być oddzielone spacjami. Nawiasy należy zapisać za pomocą znaków nawiasów okrągłych (‘(‘ oraz ‘)’). Do zapisu konkretnych operacji należy użyć odpowiednich operatorów:

* dodawanie: +,
* odejmowanie: -,
* mnożenie: \*,
* dzielenie: /,
* potęgowanie: ^.

**4. Specyfikacja wyjścia**

Program wypisuje w sposób standardowy podane wyrażenie algebraiczne w formie ONP (gdzie znaczenie operatorów jest jednakowe z opisanym powyżej; dodatkowo znak ‘~’ oznacza minus unarny, czyli zmianę znaku) oraz wynik tego wyrażenia. Jeżeli wynik jest liczbą niecałkowitą zostanie on wypisany z dokładnością sześciu cyfr po przecinku.

Wszystkie wyrażenia algebraiczne wraz z formą ONP oraz wynikiem wykonane w danym uruchomieniu programu zapisywane są w pliku *historia\_sesji.txt* i usuwane wraz z kolejnym uruchomieniem.

Jeżeli w wyrażeniu algebraicznym występuje dzielenie przez zero forma ONP nie zostanie wyświetlona i zapisana, a zamiast wyniku pojawi się komunikat: "Blad: dzielenie przez zero".

**5. Specyfikacja wewnętrzna**

Program wykorzystuje algorytm konwersji wyrażenia algebraicznego do formy ONP. Wyrażenie algebraiczne zapisane jest w postaci tablicy charów *wyrazenie*. Iterując po kolejnych elementach tablicy *wyrazenie* liczby zapisujemy do kolejnych indeksów tablicy typu double *liczby*, natomiast operatory na tablicy charów *operatory*. W trakcie obliczam wszystkie możliwe do wykonania operacje (takie, których priorytet jest wyższy od nastepnej operacji). W przypadku natrafienia na nawias zamknięty wykonywane są operacje aż do miejsca, w którym operatorem jest nawias otwarty. W przypadku gdy w wyrażeniu algebraicznym występuje dzielenie przez zero następuje wyjście z pętli, wyświetlony i zapisany zostaje komunikat "Blad: dzielenie przez zero". Jeżeli pętla została opuszczona w sposób strukturalny forma ONP oraz wynik zostaną wyświetlone oraz zapisane do pliku.

Struktura zawierająca lewą i prawą stronę operacji oraz łączący je operator:

struct op

{

double lewa;

double prawa;

char Operator;

};

Funkcje:

//nadawanie priorytetu operatorowi

int priorytet(char znak);

//obliczanie operacji

double oblicz(struct op operacja);

//wpisywanie liczby do tablic "liczby" i "onp"

void zapisywanie\_liczb();

//negacja

void zmiana\_na\_ujemne();

//zapisanie wyniku operacji w tablicy liczby i operatora w tablicy onp

void oblicz\_i\_zapisz();

Główne tablice:

char wyrazenie[200] //wyrażenie algebraiczne

onp[100] //wyrażenie algebraiczne zapisane w formie ONP

operators[100] //przechowuje kolejne operatory obecnego wyrażenia algebraicznego

double liczby[100] //przechowuje kolejne liczby obecnego wyrażenia algebraicznego