

Język programowania: Assembler (MIPS)

Zadanie:

Napisz prostą aplikację umożliwiającą wykonanie podstawowych operacji arytmetycznych (kalkulator).

Wymagania:

1. (max. 1ptk) Program musi pytać użytkownika o zdefiniowanie jakie równanie chce wykonać
2. (max. 8ptk) Program wspiera podstawowe operacje arytmetyczne:
 - dodawanie
 - odejmowanie
 - mnożenie
 - dzielenie
3. (max. 1 ptk) Program pozwala na użycie wyniku z poprzedniej operacji jako pierwszego parametru w następnej operacji

Założenia ogólne:

1. Program wyświetla stosowne komunikatu do użytkownika jeżeli są konieczne (prośba o wprowadzenie liczby, pytanie jaką operację chce wykonać, etc.)
2. Wykonywane operacje są przeprowadzanie na liczbach całkowitych

Język programowania: Assembler (MPIS)

Zadanie:

Napisz prostą aplikację umożliwiającą wykonanie podstawowych operacji arytmetycznych (kalkulator).

Wymagania:

1. (max. 1 pkt) Program musi pytać użytkownika o zdefiniowanie jakie równanie chce wykonać
2. (max. 2 pkt) Program wspiera podstawowe operacje arytmetyczne:
 - dodawanie
 - odejmowanie
 - dzielenie
 - mnożenie
 - odwrotność ($1/x$)
 - wartość bezwzględna
3. (max. 4 pkt) Program wspiera złożone operacje arytmetyczne:
 - potęgowanie
 - silnia
4. (max. 2 pkt) Program posiada bufor liczbowy (M), który umożliwia:
 - dodanie wyniku do buforu (M+)
 - odjęcie wyniku od buforu (M-)
 - zresetowanie buforu do '0' (MC)
 - wczytanie buforu jako wartości równania (MR)
5. (max. 1 pkt) Program pozwala na użycie wyniku z poprzedniej operacji jako pierwszego parametru w następnej operacji

Założenia ogólne:

1. Program wyświetla stosowne komunikatu do użytkownika jeżeli są konieczne (prośba o wprowadzenie liczby, pytanie jaką operację chce wykonać, etc.)
2. Wykonywane operacje są przeprowadzanie na liczbach rzeczywistych i całkowitych (gdzie to potrzebne)

Język programowania: Assembler (MPIS)

Generalnie nie polecam marnować czasu to zadanie to jest bardzo słaby żart

Zadanie:

Napisz prostą aplikację umożliwiającą wykonanie podstawowych równań arytmetycznych (kalkulator).

Wymagania:

1. (max. 1 pkt) Program musi pytać użytkownika o zdefiniowanie jakie równanie chce wykonać w formacie „1 + 2 * 4” wpisywane jako tekst
2. (max. 4 pkt) Program potrafi przekształcać podany teks na liczby i kolejne zmienne trzymając je w pamięci
3. (max. 2 pkt) Program wspiera podstawowe operacje arytmetyczne:
 - dodawanie
 - odejmowanie
 - dzielenie
 - mnożenie
 - odwrotność ($1/x$)
 - wartość bezwzględna
 - potęgowanie
 - silnia
4. (max. 0.5 pkt) Program posiada bufor liczbowy (M), który umożliwia:
 - dodanie wyniku do buforu (M+)
 - odjęcie wyniku od buforu (M-)
 - zresetowanie buforu do '0' (MC)
 - wczytanie buforu jako wartości równania (MR)
5. (max. 1 pkt) Program umożliwia użycie symbolu „M” jako pamięci z bufora i P jako wyniku z poprzedniego równania
6. (max. 0.5 pkt) Program pozwala na użycie wyniku z poprzedniej operacji jako pierwszego parametru w następnej operacji
7. (max. 1 pkt) Program poprawnie obsługuje kolejność operacji

Założenia ogólne:

1. Program wyświetla stosowne komunikatu do użytkownika jeżeli są konieczne (prośba o wprowadzenie równania)
2. Wykonywane operacje są przeprowadzanie na liczbach rzeczywistych i całkowitych (gdzie to potrzebne)

Punkty dodatkowe:

1. Program wspiera operacje w nawiasach (1 pkt)
2. Program obsługuje wyjątki procesora (1 pkt)

Język programowania: Assembler (MPIS)

Zadanie:

Napisz prostą aplikację umożliwiającą wykonanie podstawowych operacji arytmetycznych (kalkulator) z obsługą błędów procesora.

Wymagania:

1. (max. 2 pkt) Program musi pytać użytkownika o zdefiniowanie zmiennych oraz operacji do wykonania (lista 6)
 - o dodawanie
 - o odejmowanie
 - o dzielenie
 - o mnożenie
 - o potęgowanie
 - o silnia
2. (max. 8 pkt) Program potrafi obsłużyć min 5 z następujących błędów procesora (należy zaprezentować, w miarę możliwości naturalne wystąpienie wyjątku np. przepełnienie)
 - o ADDRESS_EXCEPTION_LOAD (4),
 - o ADDRESS_EXCEPTION_STORE (5),
 - o SYSCALL_EXCEPTION (8),
 - o BREAKPOINT_EXCEPTION (9),
 - o RESERVED_INSTRUCTION_EXCEPTION (10),
 - o ARITHMETIC_OVERFLOW_EXCEPTION (12),
 - o TRAP_EXCEPTION (13),
 - o DIVIDE_BY_ZERO_EXCEPTION (15),
 - o FLOATING_POINT_OVERFLOW (16)
 - o FLOATING_POINT_UNDERFLOW (17)
3. (max. 2 pkt extra po 0.5 za przykład instrukcji) Program potrafi obsłużyć błędy zdefiniowane przez użytkownika z użyciem odpowiednio instrukcji trap ('tge, tgeu, tgei, tgeiu' etc.)

Założenia ogólne:

1. Program wyświetla stosowne komunikaty do użytkownika jeżeli są konieczne (prośba o wprowadzenie liczby, pytanie jaką operację chce wykonać, etc.)
2. Wykonywane operacje są przeprowadzane na liczbach rzeczywistych i całkowitych (gdzie to potrzebne)