Zadanie 6.4

Wymagane

6. Dynamiczna alokacja pamięci - teksty i napisy

4. Naprawdę duże liczby II

Termin
2022-06-02
zostało 1 dzień

Trudność
Punkty: 4.5

Próba
30

Ocena maszyny
Czas testu: 100sek

Inspekcja kodu

Plagiat

Raporty
Raport główny

Napisz funkcje do sumowania, odejmowania i mnożenia dwóch dużych liczb całkowitych ze znakiem, do sprawdzania czy wyrażenie zawierające duże liczby oraz liczby są poprawne oraz do obliczania wartości wyrażenia.

Funkcje te powinny mieć następujące prototypy:

```
int multiply(const char* number1, const char* number2, char** result);
int subtract(const char* number1, const char* number2, char** result);
int add(const char* number1, const char* number2, char** result);
```

Parametry:

- `number1`, `number2` - wskaźniki na tablice znaków, zawierające liczby które mają zostać pomnożone,
- `result` - wskaźnik na wskaźnik na tablicę znaków, do której dana funkcja ma zapisać swój wynik, wykonany dla liczb `number1` oraz `number2`.

Każda funkcja musi zaalokować pamięć (dla `result`; pamiętaj o dodaniu na końcu ciągu terminatora).

Funkcje zwracają:

- `0` jeśli operacja została wykonana poprawnie
- `1` w przypadku błędnych danych wejściowych
- `2` gdy przekazane do funkcji ciągi nie są poprawnymi liczbami całkowitymi ze znakiem,
- `3` w przypadku kiedy nie udało zaalokować się pamięci na wynik.

```
int validate(const char *number);
```

Funkcja sprawdza, czy liczba przekazana w parametrze `number` jest poprawną liczbą całkowitą.

Funkcja `validate` zwraca:

- `-1` - w przypadku błędnych danych wejściowych
- `0` - jeśli ciąg znaków `number` tworzy poprawną liczbę całkowitą,
- `2` - kiedy przekazany do funkcji ciąg nie są poprawną liczbą całkowitą,

---

```
int validate_expression(const char *expr);
```

Funkcja sprawdza, czy wyrażenie przekazane w parametrze `expr` jest poprawnym wyrażeniem matematycznym.

Poprawne wyrażenie oznacza ciąg liczb rozdzielonych operatorem arytmetycznym (`+`, `-` lub `*`). Poprawne wyrażenie może zawierać tylko i wyłącznie liczby całkowite (dodatnie oraz ujemne), a więc wyrażenie postaci `5+-3` jest dopuszczalne, natomiast wyrażenie postaci `5-+3` już nie.

Funkcja zwraca:

- `0` jeśli wyrażenie jest poprawne,
- `1` w przypadku błędnego wyrażenia,
- `2` w przypadku błędnych danych wejściowych.

```
int calculate(const char* expr, char **result);
```

Funkcja oblicza wartość wyrażenia przekazanego w parametrze `expr` i zapisuje ją do zmiennej `result`. Funkcja ma obliczyć wartość `expr`, zgodnie z kolejnością występowania operatorów, a nie ich priorytetami.

Funkcja zwraca:

- `0` - jeśli udało się obliczyć wartość wyrażenia,
- `1` - w przypadku błędnych danych wejściowych
- `2` - gdy przekazane do funkcji wyrażenie nie jest poprawne,
- `3` - w przypadku kiedy nie udało zaalokować się pamięci na wynik.

Napisz program, który pobierze od użytkownika wyrażenie w postaci ciągu liczb i operacji pomiędzy nimi. Długość wyrażenia nie przekroczy 500 znaków.

Pamięć na tablicę do przechowania wyrażenia zaalokuj dynamicznie a jeżeli alokacja nie powiedzie się program powinien niezwłocznie wyświetlić komunikat `Failed to allocate memory` i zakończyć pracę z kodem błędu `8`.

Dla tak pobranego wyrażenia program ma obliczyć jego wartość, zgodnie z kolejnością występowania operatorów, a nie ich priorytetami. Dopuszczalne operatory to `+`, `-` oraz `*`. Uzyskany wynik należy wyświetlić w oddzielnej linii.

- Jeżeli alokacja pamięci na wynik programu się nie powiedzie, program powinien wyświetlić komunikat `Failed to allocate memory` i zakończyć pracę z kodem błędu `8`.
- Jeżeli wyrażenie jest nieprawidłowe to program ma poinformować o tym użytkownika komunikatem `Incorrect input` i zwrócić wartość `1`.

Przykład interakcji z programem -- sukces:

```
Podaj wyrazenie: 72+-7999--5522--9273+-568*3741*5417--6719+6664*9546--8938+-3753--4391+3988+-597*-6904-7149-5801*-8814-
-6344--4000-7424--562-4364+7300*9950+-3462*515+-965+1958*5917--2558+9242--4496--8457-4838-383+9882-695+6212⏎
22486107197960011319060544431692O2
```

Przykład interakcji z programem -- błąd w danych wejściowych:

```
Podaj wyrazenie: 366++235*275+360*581*772+39-845+143*576-747*955+577/n⏎
Incorrect input
```
