Telefony

Wygenerowano przez Doxygen 1.8.11

Rozdział 1

Dokumentacja zadania telefony

Treść zadania

Tegoroczne duże zadanie polega na zaimplementowaniu operacji na numerach telefonów. Na potrzeby tego zadania przyjmujemy, że numer telefonu jest to niepusty ciąg składający się z cyfr 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Jako pierwszą część zadania należy zaimplementować moduł operacji przekierowywania numerów telefonów.

Opis programu 1

Opis interfejsu modułu znajduje się w plikach phone_forward.h oraz phone_forward.c w formacie komentarzy dla programu doxygen.

Interfejs klasy przechowującej przekierowania numerów telefonicznych znajduje się w pliku phone_forward.h. Implementacja modułu operacji przekierowywania numerów telefonów znajduje się w pliku phone_forward.c. Przykład użycia znajduje się w pliku phone_forward_example.c.

Opis programu 2

Interfejs tekstowy

Program czyta dane ze standardowego wejścia, wyniki wypisuje na standardowe wyjście, a informacje o błędach na standardowe wyjście diagnostyczne.

Poprawne dane wejściowe

Dane wejściowe wyraża się w pewnym języku programowania. W języku tym są trzy rodzaje leksemów:

```
numer - niepusty ciąg cyfr 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;
identyfikator - niepusty ciąg składający się z małych lub dużych liter alfabetu angielskiego i cyfr dziesiętny
operator.
```

W języku tym są cztery operatory:

```
NEW
DEL
>
?
```

Słowa NEW i DEL są zastrzeżone – nie może być takich identyfikatorów.

Język udostępnia następujące operacje tworzenia, przełączania i usuwania bazy przekierowań:

```
NEW identyfikator - jeśli baza przekierowań o podanej nazwie nie istnieje, to tworzy nową bazę o tej nazwie i
DEL identyfikator - usuwa bazę przekierowań o podanej nazwie;
```

Język udostępnia następujące operacje, dotyczące aktualnej bazy przekierowań, wykonywane na numerach:

```
numer > numer - dodaje przekierowanie numerów;
numer ? - wypisuje przekierowanie z podanego numeru;
? numer - wypisuje przekierowania na podany numer;
DEL numer - usuwa wszystkie przekierowania, których numer jest prefiksem.
```

Między leksemami może nie być odstępu albo może być dowolna liczba białych znaków (spacja, tabulator, znak nowej linii, znak powrotu karetki). Między leksemami musi być co najmniej jeden biały znak, jeśli jego brak powodowałby błędną interpretację.

W języku mogą pojawić się komentarze. Komentarz rozpoczyna i kończy się sekwencją $$.

Skrypt

Skrypt dla podanego numeru y wyznaczy wszystkie takie numery x, że phfwdGet(x) = y. Skrypt przyjmuje trzy parametry:

```
pierwszy wskazuje (ścieżka i nazwa) na plik wykonywalny programu, który jest wyspecyfikowany w poprzednim punk
drugi wskazuje (ścieżka i nazwa) na plik z operacjami przekierowania numerów, zawiera ciąg użyć operatora >;
trzeci to numer y.
```

Rozdział 2

Indeks struktur danych

2.1 Struktury danych

Tutaj znajdują się struktury danych wraz z ich krótkimi opisami:

Rozdział 3

Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

Rozdział 4

Dokumentacja struktur danych

4.1 Dokumentacja struktury BaseNode

Struktura przechowująca bazy przekierowań.

`#include <base.h>`

Diagram współpracy dla BaseNode:

Pola danych

- char * id

  identyfikator bazy przekierowań
- struct PhoneForward * pf

  wskaźnik na bazę przekierowań
- struct BaseNode * left

  wskaźnik na lewego syna; poddrzewo zawierające przekierowania o identyfikatorze mniejszym wg porządku leksykograficznego
- struct BaseNode * right

  wskaźnik na prawego syna; poddrzewo zawierające przekierowania o identyfikatorze większym wg porządku leksykograficznego

4.1.1 Opis szczegółowy

Struktura przechowująca bazy przekierowań.

Struktura drzewiasta - wierzchołek reprezentuje bazę oraz jej identyfikator.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- base.h

4.2 Dokumentacja struktury Num

Struktura przechowująca numer telefonu.

`#include <utils.h>`

Pola danych

- char * array
  wskaźnik na tablicę przechowującą znaki napisu
- int size
  rozmiar zaalokowanej tablicy
- int taken
  liczba zajętych komórek tablicy

4.2.1 Opis szczegółowy

Struktura przechowująca numer telefonu.

Struktura stworzona na potrzeby modyfikowania na bieżąco numeru telefonu.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- utils.h

4.3 Dokumentacja struktury PhoneForward

Struktura przechowująca przekierowania numerów telefonów.

`#include <phone_forward.h>`

Diagram współpracy dla PhoneForward:

Pola danych

- struct PhoneForwardTrieFromTo * trieFromTo
  drzewo przekierowań
- struct PhoneForwardTrieToFrom * trieToFrom
  drzewo odwrotności przekierowań

4.3.1 Opis szczegółowy

Struktura przechowująca przekierowania numerów telefonów.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- phone_forward.h

4.4 Dokumentacja struktury PhoneForwardTrieFromTo

Struktura drzewiasta do przechowywania numerów zgodnie z prefiksami oraz ich przekierowań.

`#include <utils.h>`

Diagram współpracy dla PhoneForwardTrieFromTo:

Pola danych

- char * number
  prefiks, na który przekierowujemy numer reprezentowany przez wierzchołek
- struct PhoneForwardTrieFromTo * children [NUMBER_OF_DIGITS]
  tablica synów wierzchołka

4.4.1 Opis szczegółowy

Struktura drzewiasta do przechowywania numerów zgodnie z prefiksami oraz ich przekierowań.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- utils.h

4.5 Dokumentacja struktury PhoneForwardTrieToFrom

Struktura drzewiasta do przechowywania numerów zgodnie z prefiksami oraz przekierowań na dany numer.

`#include <utils.h>`

Diagram współpracy dla PhoneForwardTrieToFrom:

Pola danych

- struct PhoneNumbers * phoneNumbers
  prefiksy, które zostały przekierowane na numer reprezentowany przez wierzchołek
- struct PhoneForwardTrieToFrom * children [NUMBER_OF_DIGITS]
  tablica synów wierzchołka

4.5.1 Opis szczegółowy

Struktura drzewiasta do przechowywania numerów zgodnie z prefiksami oraz przekierowań na dany numer.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- utils.h

4.6 Dokumentacja struktury PhoneNumbers

Struktura przechowująca ciąg numerów telefonów.

`#include <phone_forward.h>`

Pola danych

- char ** array
  wskaźnik na wskaźnik na tablicę napisów reprezentujących numery telefonów
- int arraySize
  rozmiar zaalokowanej tablicy
- int numberOfPhoneNumbers
  liczba komórek tablicy zajętych przez napisy

4.6.1 Opis szczegółowy

Struktura przechowująca ciąg numerów telefonów.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- phone_forward.h

4.7 Dokumentacja struktury WorkingStruct

Struktura robocza przechowująca elementy potrzebne do czytania zdefiniowanego w zadaniu języka programowania.

`#include <base.h>`

Diagram współpracy dla WorkingStruct:

Pola danych

- struct PhoneForward * pf
  wskaźnik na aktualną bazę przekierowań
- struct BaseNode * bases
  wskaźnik na strukturę drzewiastą przechowującą bazy przekierowań
- struct Num * argument1
  struktura reprezentująca napis do przechowywania argumentu funkcji - identyfikatora bazy lub numeru telefonu
- struct Num * argument2
  struktura reprezentująca napis do przechowywania drugiego argumentu funkcji - identyfikatora bazy lub numeru telefonu

4.7.1 Opis szczegółowy

Struktura robocza przechowująca elementy potrzebne do czytania zdefiniowanego w zadaniu języka programowania.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- base.h

Rozdział 5

Dokumentacja plików

5.1 Dokumentacja pliku base.c

Implementacja interfejsu klasy przechowującej bazy przekierowań

```
#include <stdlib.h>
#include <string.h>
#include "utils.h"
#include "base.h"
```

Wykres zależności załączania dla base.c:

5.2 Dokumentacja pliku base.h

Interfejs klasy przechowującej bazy przekierowań

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdbool.h>
#include "utils.h"
```

Wykres zależności załączania dla base.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Struktury danych

- struct BaseNode

  Struktura przechowująca bazy przekierowań.
- struct WorkingStruct

  Struktura robocza przechowująca elementy potrzebne do czytania zdefiniowanego w zadaniu języka programowania.

Funkcje

- struct WorkingStruct * newWorkingStruct ()

  Tworzy strukturę.
- void delNode (struct BaseNode *n)

  Usuwa strukturę.
- struct BaseNode * newBase (struct BaseNode *n, char *id, struct PhoneForward **pf)

  Dodaje bazę.
- struct BaseNode * delBase (struct BaseNode *n, char *id, struct PhoneForward **pf, bool *correct)

  Usuwa wierzchołek.
- struct BaseNode * findMinimumOnTheRight (struct BaseNode *n)

  Znajduje wierzchołek.

5.2.1 Opis szczegółowy

Interfejs klasy przechowującej bazy przekierowań

Autor

Magdalena Augustyńska ma370723@mimuw.edu.pl

Data

01.06.2018

5.2.2 Dokumentacja funkcji

5.2.2.1 struct BaseNode* delBase (struct BaseNode * *n*, char * *id*, struct PhoneForward ** *pf*, bool * *correct*)

Usuwa wierzchołek.

Usuwa z drzewa wierzchołek o podanym identyfikatorze.

Parametry

n	- wskaźnik na korzeń drzewa przechowującego bazy przekierowań;
id	- napis reprezentujący identyfikator bazy, którą usuwamy;
pf	- wskaźnik na wskaźnik na aktualną bazę przekierowań;
correct	- parametr informujący o tym, czy próbowano usunąć bazę nieobecną w drzewie - modyfikowany w takim przypadku.

Zwraca

Wskaźnik na korzeń zmodyfikowanego drzewa.

Oto graf wywołań dla tej funkcji:

5.2.2.2 void delNode (struct BaseNode * n)

Usuwa strukturę.

Usuwa strukturę drzewiastą przechowującą bazy przekierowań.

Parametry

n	- wskaźnik na strukturę do usunięcia.

Oto graf wywołań dla tej funkcji:

5.2.2.3 struct BaseNode* findMinimumOnTheRight (struct BaseNode * n)

Znajduje wierzchołek.

Pomocnicza funkcja do usuwania wierzchołka z drzewa baz przekierowań. Znajduje wierzchołek, który jest wstawiany w miejsce usuwanego i usuwa krawędź między nim a rodzicem.

Parametry

n	- wskaźnik na korzeń przeszukiwanego poddrzewa.

Zwraca

Wskaźnik na znaleziony wierzchołek drzewa

5.2.2.4 struct BaseNode* newBase (struct BaseNode * n, char * id, struct PhoneForward ** pf)

Dodaje bazę.

Dodaje wierzchołek reprezentujący bazę przekierowań do struktury drzewiastej.

Parametry

n	- wskaźnik na korzeń drzewa przechowującego bazy przekierowań;
id	- napis reprezentujący identyfikator bazy, którą dodajemy;
pf	- wskaźnik na wskaźnik na aktualną bazę przekierowań;

Zwraca

Wskaźnik na korzeń zmodyfikowanego drzewa.

Oto graf wywołań dla tej funkcji:

5.2.2.5 struct WorkingStruct* newWorkingStruct ( )

Tworzy strukturę.

Tworzy nową strukturę roboczą zawierającą niezainicjalizowane struktury.

Zwraca

Wskaźnik na utworzoną strukturę.

5.3 Dokumentacja pliku main.c

Funkcja main modułu udostępniającego operacje na numerach telefonów przez interfejs tekstowy.

```
#include <stdlib.h>
#include "base.h"
#include "parser.h"
```

Wykres zależności załączania dla main.c:

Funkcje

- int main ()

  Przeprowadza program.

5.3.1 Opis szczegółowy

Funkcja main modułu udostępniającego operacje na numerach telefonów przez interfejs tekstowy.

Autor

Magdalena Augustyńska ma370723@mimuw.edu.pl

Data

01.06.2018

5.3.2 Dokumentacja funkcji

5.3.2.1 int main ( )

Przeprowadza program.

Czyta dane ze standardowego wejścia, wyniki wypisuje na standardowe wyjście, a informacje o błędach na standardowe wyjście diagnostyczne.

Zwraca

Wartość 0, jeśli program przetworzył wszystkie dane wejściowe i zakończył się poprawnie. W przeciwnym przypadku kończy działanie programu i nic nie zwraca.

Oto graf wywołań dla tej funkcji:

5.4 Dokumentacja pliku parser.c

Implementacja modułu udostępniającego operacje na numerach telefonów przez interfejs tekstowy.

```
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#include "phone_forward.h"
#include "base.h"
#include "utils.h"
```

Wykres zależności załączania dla parser.c:

Funkcje

- void freeMemory (struct WorkingStruct ∗ws)

  Zwalnia pamięć.
- void end (struct WorkingStruct ∗ws)

  Kończy działanie programu.
- void syntaxError (int signNumber, struct WorkingStruct ∗ws)

  Obsługuje błąd składniowy.
- void executionError (int signNumber, char ∗op, struct WorkingStruct ∗ws)

  Obsługuje błąd wykonania.
- void eofError (struct WorkingStruct ∗ws)

  Obsługuje błąd końca danych.
- void printPhfwdReverse (int signNumber, struct WorkingStruct ∗ws)

  Wypisuje przekierowania.
- void printPhfwdGet (int signNumber, struct WorkingStruct ∗ws)

  Wypisuje przekierowania.
- void add (int signNumber, struct WorkingStruct ∗ws)

  Dodaje przekierowanie.
- bool correctId (char ∗c)

  Sprawdza poprawność identyfikatora.
- int parse (struct WorkingStruct ∗ws)

  Czyta wejście.

5.4.1 Opis szczegółowy

Implementacja modułu udostępniającego operacje na numerach telefonów przez interfejs tekstowy.

Autor

Magdalena Augustyńska ma370723@mimuw.edu.pl

Data

01.06.2018

5.4.2 Dokumentacja funkcji

5.4.2.1 void add (int *signNumber,* struct WorkingStruct ∗ *ws*)

Dodaje przekierowanie.

Dodaje przekierowanie z numeru `argument1` na numer `argument2` znajdujących się w strukturze roboczej, do bazy wskazywanej przez `pf`.

Parametry

signNumber	- numer znaku operatora ">";
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.4.2.2 bool correctId (char * c)

Sprawdza poprawność identyfikatora.

Sprawdza, czy identyfikator nie jest jednym z zastrzeżonych napisów.

Parametry

c	- wskaźnik na napis.

Zwraca

Wartość `true`, jeśli napis nie jest zastrzeżonym identyfikatorem. Wartość `false` w przeciwnym przypadku.

5.4.2.3 void end (struct WorkingStruct * ws)

Kończy działanie programu.

Zwalnia zaalokowaną pamięć i kończy działanie programu z błędem.

Parametry

ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.4.2.4 void eofError (struct WorkingStruct * ws)

Obsługuje błąd końca danych.

Wypisuje komunikat o niespodziewanym końcu danych wejściowych na standardowe wyjście diagnostyczne i kończy działanie programu.

Parametry

ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.4.2.5 void executionError (int *signNumber,* char * *op,* struct WorkingStruct * *ws*)

Obsługuje błąd wykonania.

Wypisuje komunikat o błędzie wykonania na standardowe wyjście diagnostyczne i kończy działanie programu.

Parametry

signNumber	- numer pierwszego znaku operatora, którego wykonanie spowodowało błąd;
op	- napis reprezentujący nazwę operatora, którego wykonanie spowodowało błąd;
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.4.2.6 void freeMemory (struct WorkingStruct * *ws*)

Zwalnia pamięć.

Zwalnia pamięć zaalokowaną podczas działania programu.

Parametry

ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.4.2.7 int parse (struct WorkingStruct * *ws*)

Czyta wejście.

Czyta dane ze standardowego wejścia, wyniki wypisuje na standardowe wyjście, a informacje o błędach na standardowe wyjście diagnostyczne.

Parametry

ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu;

Zwraca

Numer stanu, w którym program zakończył działanie.

Oto graf wywołań dla tej funkcji:

5.4.2.8 void printPhfwdGet (int *signNumber,* struct WorkingStruct * *ws*)

Wypisuje przekierowania.

Wypisuje przekierowanie z numeru znajdującego się w strukturze roboczej przekazanej jako argument. Numer ten jest wskazywany przez wskaźnik `argument1`. Aktualna baza przekierowań jest wskazywana przez wskaźnik `pf` w strukturze roboczej.

Parametry

signNumber	- numer znaku operatora "?";
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.4.2.9 void printPhfwdReverse (int *signNumber,* struct WorkingStruct * *ws*)

Wypisuje przekierowania.

Wypisuje przekierowanie na numer znajdujący się w strukturze roboczej przekazanej jako argument. Numer ten jest wskazywany przez wskaźnik `argument1`. Aktualna baza przekierowań jest wskazywana przez wskaźnik `pf` w strukturze roboczej.

Parametry

signNumber	- numer znaku operatora "?";
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.4.2.10 void syntaxError (int *signNumber,* struct WorkingStruct * *ws*)

Obsługuje błąd składniowy.

Wypisuje komunikat o błędzie składniowym na standardowe wyjście diagnostyczne i kończy działanie programu.

Parametry

signNumber	- numer pierwszego znaku, który nie daje się zinterpretować jako poprawne wejście;
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.5 Dokumentacja pliku parser.h

Interfejs modułu udostępniającego operacje na numerach telefonów przez interfejs tekstowy.

```
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#include "phone_forward.h"
#include "base.h"
```

Wykres zależności załączania dla parser.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Funkcje

- void freeMemory (struct WorkingStruct ∗ws)

  Zwalnia pamięć.
- void end (struct WorkingStruct ∗ws)

  Kończy działanie programu.
- void syntaxError (int signNumber, struct WorkingStruct ∗ws)

  Obsługuje błąd składniowy.
- void executionError (int signNumber, char ∗op, struct WorkingStruct ∗ws)

  Obsługuje błąd wykonania.
- void eofError (struct WorkingStruct ∗ws)

  Obsługuje błąd końca danych.
- void printPhfwdReverse (int signNumber, struct WorkingStruct ∗ws)

  Wypisuje przekierowania.
- void printPhfwdGet (int signNumber, struct WorkingStruct ∗ws)

  Wypisuje przekierowania.
- void add (int signNumber, struct WorkingStruct ∗ws)

  Dodaje przekierowanie.
- bool correctId (char ∗c)

  Sprawdza poprawność identyfikatora.
- int parse (struct WorkingStruct ∗ws)

  Czyta wejście.

5.5.1 Opis szczegółowy

Interfejs modułu udostępniającego operacje na numerach telefonów przez interfejs tekstowy.

Autor

Magdalena Augustyńska `ma370723@mimuw.edu.pl`

Data

01.06.2018

5.5.2 Dokumentacja funkcji

5.5.2.1 void add (int *signNumber,* struct WorkingStruct ∗ *ws*)

Dodaje przekierowanie.

Dodaje przekierowanie z numeru `argument1` na numer `argument2` znajdujących się w strukturze roboczej, do bazy wskazywanej przez `pf`.

Parametry

signNumber	- numer znaku operatora ">";
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.5.2.2 bool correctId (char * c)

Sprawdza poprawność identyfikatora.

Sprawdza, czy identyfikator nie jest jednym z zastrzeżonych napisów.

Parametry

c	- wskaźnik na napis.

Zwraca

Wartość `true`, jeśli napis nie jest zastrzeżonym identyfikatorem. Wartość `false` w przeciwnym przypadku.

5.5.2.3 void end (struct WorkingStruct * ws)

Kończy działanie programu.

Zwalnia zaalokowaną pamięć i kończy działanie programu z błędem.

Parametry

ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.5.2.4 void eofError (struct WorkingStruct * ws)

Obsługuje błąd końca danych.

Wypisuje komunikat o niespodziewanym końcu danych wejściowych na standardowe wyjście diagnostyczne i kończy działanie programu.

Parametry

ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.5.2.5 void executionError (int *signNumber,* char * *op,* struct WorkingStruct * *ws*)

Obsługuje błąd wykonania.

Wypisuje komunikat o błędzie wykonania na standardowe wyjście diagnostyczne i kończy działanie programu.

Parametry

signNumber	- numer pierwszego znaku operatora, którego wykonanie spowodowało błąd;
op	- napis reprezentujący nazwę operatora, którego wykonanie spowodowało błąd;
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.5.2.6 void freeMemory (struct WorkingStruct * *ws*)

Zwalnia pamięć.

Zwalnia pamięć zaalokowaną podczas działania programu.

Parametry

ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.5.2.7 int parse (struct WorkingStruct * *ws*)

Czyta wejście.

Czyta dane ze standardowego wejścia, wyniki wypisuje na standardowe wyjście, a informacje o błędach na standardowe wyjście diagnostyczne.

Parametry

ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu;

Zwraca

Numer stanu, w którym program zakończył działanie.

Oto graf wywołań dla tej funkcji:

5.5.2.8 void printPhfwdGet (int *signNumber,* struct WorkingStruct * *ws*)

Wypisuje przekierowania.

Wypisuje przekierowanie z numeru znajdującego się w strukturze roboczej przekazanej jako argument. Numer ten jest wskazywany przez wskaźnik `argument1`. Aktualna baza przekierowań jest wskazywana przez wskaźnik `pf` w strukturze roboczej.

Parametry

signNumber	- numer znaku operatora "?";
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.5.2.9 void printPhfwdReverse (int *signNumber,* struct WorkingStruct * *ws*)

Wypisuje przekierowania.

Wypisuje przekierowanie na numer znajdujący się w strukturze roboczej przekazanej jako argument. Numer ten jest wskazywany przez wskaźnik `argument1`. Aktualna baza przekierowań jest wskazywana przez wskaźnik `pf` w strukturze roboczej.

Parametry

signNumber	- numer znaku operatora "?";
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.5.2.10 void syntaxError (int *signNumber,* struct WorkingStruct * *ws*)

Obsługuje błąd składniowy.

Wypisuje komunikat o błędzie składniowym na standardowe wyjście diagnostyczne i kończy działanie programu.

Parametry

signNumber	- numer pierwszego znaku, który nie daje się zinterpretować jako poprawne wejście;
ws	- wskaźnik na strukturę roboczą, w której znajdują się wszystkie struktury w użyciu podczas działania programu.

Oto graf wywołań dla tej funkcji:

5.6 Dokumentacja pliku phone_forward.c

Implementacja modułu operacji przekierowywania numerów telefonów.

```
#include <stdbool.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "phone_forward.h"
#include "utils.h"
```

Wykres zależności załączania dla phone_forward.c:

Funkcje

- struct PhoneForward ∗ phfwdNew (void)
  Tworzy nową strukturę.
- void phfwdDelete (struct PhoneForward ∗pf)
  Usuwa strukturę.
- bool phfwdAdd (struct PhoneForward ∗pf, char const ∗num1, char const ∗num2)
  Dodaje przekierowanie.
- void phfwdRemove (struct PhoneForward ∗pf, char const ∗num)
  Usuwa przekierowania.
- struct PhoneNumbers const ∗ phfwdGet (struct PhoneForward ∗pf, char const ∗num)
  Wyznacza przekierowanie numeru.
- struct PhoneNumbers const ∗ phfwdReverse (struct PhoneForward ∗pf, char const ∗num)
  Wyznacza przekierowania na dany numer.
- char const ∗ phnumGet (struct PhoneNumbers const ∗pnum, size_t idx)
  Udostępnia numer.
- size_t **phfwdNonTrivialCount** (struct PhoneForward ∗pf, char const ∗set, size_t len)

5.6.1 Opis szczegółowy

Implementacja modułu operacji przekierowywania numerów telefonów.

Autor

Magdalena Augustyńska ma370723@mimuw.edu.pl

Data

01.06.2018

5.6.2 Dokumentacja funkcji

5.6.2.1 bool phfwdAdd (struct PhoneForward ∗ *pf,* char const ∗ *num1,* char const ∗ *num2*)

Dodaje przekierowanie.

Dodaje przekierowanie wszystkich numerów mających prefiks `num1`, na numery, w których ten prefiks zamieniono odpowiednio na prefiks `num2`. Każdy numer jest swoim własnym prefiksem. Jeśli wcześniej zostało dodane przekierowanie z takim samym parametrem `num1`, to jest ono zastępowane.

Parametry

in	*pf*	– wskaźnik na strukturę przechowującą przekierowania numerów;
in	*num1*	– wskaźnik na napis reprezentujący prefiks numerów przekierowywanych;
in	*num2*	– wskaźnik na napis reprezentujący prefiks numerów, na które jest wykonywane przekierowanie.

Zwraca

Wartość `true`, jeśli przekierowanie zostało dodane. Wartość `false`, jeśli wystąpił błąd, np. podany napis nie reprezentuje numeru, oba podane numery są identyczne lub nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.6.2.2 void phfwdDelete (struct PhoneForward * *pf*)

Usuwa strukturę.

Usuwa strukturę wskazywaną przez `pf`. Nic nie robi, jeśli wskaźnik ten ma wartość NULL.

Parametry

in	*pf*	– wskaźnik na usuwaną strukturę.

Oto graf wywołań dla tej funkcji:

5.6.2.3 struct PhoneNumbers const* phfwdGet (struct PhoneForward * *pf*, char const * *num*)

Wyznacza przekierowanie numeru.

Wyznacza przekierowanie podanego numeru. Szuka najdłuższego pasującego prefiksu. Wynikiem jest co najwyżej jeden numer. Jeśli dany numer nie został przekierowany, to wynikiem jest ten numer. Jeśli podany napis nie reprezentuje numeru, wynikiem jest pusty ciąg. Alokuje strukturę `PhoneNumbers`, która musi być zwolniona za pomocą funkcji phnumDelete.

Parametry

in	*pf*	– wskaźnik na strukturę przechowującą przekierowania numerów;
in	*num*	– wskaźnik na napis reprezentujący numer.

Zwraca

Wskaźnik na strukturę przechowującą ciąg numerów lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.6.2.4 struct PhoneForward* phfwdNew (void)

Tworzy nową strukturę.

Tworzy nową strukturę niezawierającą żadnych przekierowań.

Zwraca

Wskaźnik na utworzoną strukturę lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.6.2.5 void phfwdRemove (struct PhoneForward * *pf*, char const * *num*)

Usuwa przekierowania.

Usuwa wszystkie przekierowania, w których parametr `num` jest prefiksem parametru `num1` użytego przy dodawaniu. Jeśli nie ma takich przekierowań lub napis nie reprezentuje numeru, nic nie robi.

Parametry

in	*pf*	– wskaźnik na strukturę przechowującą przekierowania numerów;
in	*num*	– wskaźnik na napis reprezentujący prefiks numerów.

Oto graf wywołań dla tej funkcji:

5.6.2.6 struct PhoneNumbers const* phfwdReverse (struct PhoneForward * *pf*, char const * *num*)

Wyznacza przekierowania na dany numer.

Wyznacza wszystkie przekierowania na podany numer. Wynikowy ciąg zawiera też dany numer. Wynikowe numery są posortowane leksykograficznie i nie mogą się powtarzać. Jeśli podany napis nie reprezentuje numeru, wynikiem jest pusty ciąg. Alokuje strukturę `PhoneNumbers`, która musi być zwolniona za pomocą funkcji phnumDelete.

Parametry

in	*pf*	– wskaźnik na strukturę przechowującą przekierowania numerów;
in	*num*	– wskaźnik na napis reprezentujący numer.

Zwraca

Wskaźnik na strukturę przechowującą ciąg numerów lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.6.2.7 char const* phnumGet (struct PhoneNumbers const * *pnum*, size_t *idx*)

Udostępnia numer.

Udostępnia wskaźnik na napis reprezentujący numer. Napisy są indeksowane kolejno od zera.

Parametry

in	*pnum*	– wskaźnik na strukturę przechowującą ciąg napisów;
in	*idx*	– indeks napisu.

Zwraca

Wskaźnik na napis. Wartość NULL, jeśli wskaźnik `pnum` ma wartość NULL lub indeks ma za dużą wartość.

Oto graf wywołań dla tej funkcji:

5.7 Dokumentacja pliku phone_forward.h

Interfejs klasy przechowującej przekierowania numerów telefonicznych.

```
#include <stdbool.h>
#include <stddef.h>
#include <stdlib.h>
```

Wykres zależności załączania dla phone_forward.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Struktury danych

- struct PhoneForward

  Struktura przechowująca przekierowania numerów telefonów.
- struct PhoneNumbers

  Struktura przechowująca ciąg numerów telefonów.

Definicje

- #define NUMBER_OF_DIGITS 10

  Liczba różnych cyfr.
- #define INIT_SIZE 5

  Początkowy rozmiar tablicy w strukturach zawierających tablicę.
- #define MULTIPLIER 2

  Mnożnik użyty do dynamicznej alokacji pamięci.

Funkcje

- struct PhoneForward ∗ phfwdNew (void)

  Tworzy nową strukturę.
- void phfwdDelete (struct PhoneForward ∗pf)

  Usuwa strukturę.
- bool phfwdAdd (struct PhoneForward ∗pf, char const ∗num1, char const ∗num2)

  Dodaje przekierowanie.
- void phfwdRemove (struct PhoneForward ∗pf, char const ∗num)

  Usuwa przekierowania.
- struct PhoneNumbers const ∗ phfwdGet (struct PhoneForward ∗pf, char const ∗num)

  Wyznacza przekierowanie numeru.
- struct PhoneNumbers const ∗ phfwdReverse (struct PhoneForward ∗pf, char const ∗num)

  Wyznacza przekierowania na dany numer.
- static void phnumDelete (struct PhoneNumbers const ∗pnum)

  Usuwa strukturę.
- char const ∗ phnumGet (struct PhoneNumbers const ∗pnum, size_t idx)

  Udostępnia numer.
- size_t **phfwdNonTrivialCount** (struct PhoneForward ∗pf, char const ∗set, size_t len)

5.7.1 Opis szczegółowy

Interfejs klasy przechowującej przekierowania numerów telefonicznych.

Autor

Magdalena Augustyńska ma370723@mimuw.edu.pl

Copyright

Uniwersytet Warszawski

Data

01.06.2018

5.7.2 Dokumentacja funkcji

5.7.2.1 bool phfwdAdd (struct PhoneForward * *pf*, char const * *num1*, char const * *num2*)

Dodaje przekierowanie.

Dodaje przekierowanie wszystkich numerów mających prefiks `num1`, na numery, w których ten prefiks zamieniono odpowiednio na prefiks `num2`. Każdy numer jest swoim własnym prefiksem. Jeśli wcześniej zostało dodane przekierowanie z takim samym parametrem `num1`, to jest ono zastępowane.

Parametry

in	*pf*	– wskaźnik na strukturę przechowującą przekierowania numerów;
in	*num1*	– wskaźnik na napis reprezentujący prefiks numerów przekierowywanych;
in	*num2*	– wskaźnik na napis reprezentujący prefiks numerów, na które jest wykonywane przekierowanie.

Zwraca

Wartość `true`, jeśli przekierowanie zostało dodane. Wartość `false`, jeśli wystąpił błąd, np. podany napis nie reprezentuje numeru, oba podane numery są identyczne lub nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.7.2.2 void phfwdDelete (struct PhoneForward * *pf*)

Usuwa strukturę.

Usuwa strukturę wskazywaną przez `pf`. Nic nie robi, jeśli wskaźnik ten ma wartość NULL.

Parametry

in	*pf*	– wskaźnik na usuwaną strukturę.

Oto graf wywołań dla tej funkcji:

5.7.2.3 struct PhoneNumbers const* phfwdGet (struct PhoneForward * *pf*, char const * *num*)

Wyznacza przekierowanie numeru.

Wyznacza przekierowanie podanego numeru. Szuka najdłuższego pasującego prefiksu. Wynikiem jest co najwyżej jeden numer. Jeśli dany numer nie został przekierowany, to wynikiem jest ten numer. Jeśli podany napis nie reprezentuje numeru, wynikiem jest pusty ciąg. Alokuje strukturę `PhoneNumbers`, która musi być zwolniona za pomocą funkcji phnumDelete.

Parametry

`in`	*pf*	– wskaźnik na strukturę przechowującą przekierowania numerów;
`in`	*num*	– wskaźnik na napis reprezentujący numer.

Zwraca

Wskaźnik na strukturę przechowującą ciąg numerów lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.7.2.4 struct PhoneForward* phfwdNew (void)

Tworzy nową strukturę.

Tworzy nową strukturę niezawierającą żadnych przekierowań.

Zwraca

Wskaźnik na utworzoną strukturę lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.7.2.5 void phfwdRemove (struct PhoneForward * *pf*, char const * *num*)

Usuwa przekierowania.

Usuwa wszystkie przekierowania, w których parametr `num` jest prefiksem parametru `num1` użytego przy dodawaniu. Jeśli nie ma takich przekierowań lub napis nie reprezentuje numeru, nic nie robi.

Parametry

`in`	*pf*	– wskaźnik na strukturę przechowującą przekierowania numerów;
`in`	*num*	– wskaźnik na napis reprezentujący prefiks numerów.

Oto graf wywołań dla tej funkcji:

5.7.2.6 struct PhoneNumbers const* phfwdReverse (struct PhoneForward * *pf*, char const * *num*)

Wyznacza przekierowania na dany numer.

Wyznacza wszystkie przekierowania na podany numer. Wynikowy ciąg zawiera też dany numer. Wynikowe numery są posortowane leksykograficznie i nie mogą się powtarzać. Jeśli podany napis nie reprezentuje numeru, wynikiem jest pusty ciąg. Alokuje strukturę `PhoneNumbers`, która musi być zwolniona za pomocą funkcji phnumDelete.

Parametry

in	*pf*	– wskaźnik na strukturę przechowującą przekierowania numerów;
in	*num*	– wskaźnik na napis reprezentujący numer.

Zwraca

Wskaźnik na strukturę przechowującą ciąg numerów lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.7.2.7 static void phnumDelete (struct PhoneNumbers const * *pnum*) [inline], [static]

Usuwa strukturę.

Usuwa strukturę wskazywaną przez `pnum`. Nic nie robi, jeśli wskaźnik ten ma wartość NULL.

Parametry

in	*pnum*	– wskaźnik na usuwaną strukturę.

Oto graf wywołań dla tej funkcji:

5.7.2.8 char const* phnumGet (struct PhoneNumbers const * *pnum*, size_t *idx*)

Udostępnia numer.

Udostępnia wskaźnik na napis reprezentujący numer. Napisy są indeksowane kolejno od zera.

Parametry

in	*pnum*	– wskaźnik na strukturę przechowującą ciąg napisów;
in	*idx*	– indeks napisu.

Zwraca

Wskaźnik na napis. Wartość NULL, jeśli wskaźnik `pnum` ma wartość NULL lub indeks ma za dużą wartość.

Oto graf wywołań dla tej funkcji:

5.8 Dokumentacja pliku utils.c

Implementacja interfejsu funkcji pomocniczych dla modułu operacji przekierowywania numerów telefonów.

```
#include <stdbool.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "phone_forward.h"
#include "utils.h"
```

Wykres zależności załączania dla utils.c:

Funkcje

- char * strdup (const char *p)

  Tworzy kopię napisu.
- struct PhoneForwardTrieFromTo * phfwdtrieFromToNew (void)

  Tworzy nową strukturę.
- struct PhoneNumbers * phnumNew (void)

  Tworzy nową strukturę.
- struct PhoneForwardTrieToFrom * phfwdtrieToFromNew (void)

  Tworzy nową strukturę.
- int **isDigit** (char c)
- bool onlyDigits (const char *s)

  Sprawdza, czy napis jest numerem telefonu.
- void phfwdTrieToFromDelete (struct PhoneForwardTrieToFrom **pft)

  Usuwa strukturę.
- void phfwdTrieFromToDelete (struct PhoneForwardTrieFromTo **pft)

  Usuwa strukturę.
- bool phnumAdd (struct PhoneNumbers *pnum, char const *num)

  Dodaje numer.
- int idxInPhNum (struct PhoneNumbers *pnum, const char *num)

  Szuka numeru.
- void onePhfwdRemove (struct PhoneForwardTrieToFrom *pft, const char *num1, const char *num2)

  Usuwa przekierowanie.
- struct Num * newNum ()

  Tworzy strukturę.
- void addToNum (struct Num *n, char add)

  Dodaje znak.
- void cutLast (struct Num *n)

  Usuwa znak.
- void delNum (struct Num *n)

  Usuwa strukturę.
- void traverse (struct PhoneForwardTrieFromTo *tmp1, struct PhoneForwardTrieToFrom *tmp2, struct Num *n)

  Usuwa przekierowania.
- bool phnumAddLexic (struct PhoneNumbers *pnum, char *num)

  Dodaje numer.

5.8.1 Opis szczegółowy

Implementacja interfejsu funkcji pomocniczych dla modułu operacji przekierowywania numerów telefonów.

Autor

Magdalena Augustyńska ma370723@mimuw.edu.pl

Data

01.06.2018

5.8.2 Dokumentacja funkcji

5.8.2.1 void addToNum (struct Num * n, char add)

Dodaje znak.

Modyfikuje napis przechowywany przez strukturę dodając znak na końcu napisu.

Parametry

n	- wskaźnik na strukturę przechowującą napis;
add	- znak, który ma być dodany.

Oto graf wywołań dla tej funkcji:

5.8.2.2 void cutLast (struct Num * n)

Usuwa znak.

Usuwa ostatni znak z napisu reprezentowangeo przez strukturę.

Parametry

n	- wskaźnik na strukturę przechowującą napis.

5.8.2.3 void delNum (struct Num * n)

Usuwa strukturę.

Usuwa strukturę przechowującą numer telefonu.

Parametry

n	- wskaźnik na strukturę przechowującą numer telefonu.

5.8.2.4 int idxInPhNum (struct PhoneNumbers * *pnum*, const char * *num*)

Szuka numeru.

Szuka indeksu, pod którym znajduje się numer telefonu w tablicy przechowywanej przez strukturę.

Parametry

pnum	- wskaźnik na strukturę przechowującą numery telefonów;
num	- wskaźnik na napis reprezentujący numer telefonu.

Zwraca

Liczba całkowita reprezentująca indeks w tablicy, pod którym znajduje się numer lub -1, gdy nie znaleziono podanego numeru.

5.8.2.5 struct Num* newNum ( )

Tworzy strukturę.

Tworzy nową strukturę reprezentującą numer telefonu.

Zwraca

Wskaźnik na strukturę reprezentującą numer lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.8.2.6 void onePhfwdRemove (struct PhoneForwardTrieToFrom * *pft*, const char * *num1*, const char * *num2*)

Usuwa przekierowanie.

Usuwa przekierowanie z drzewa odwrotności przekierowań wszystkich numerów mających prefiks `num2` na numery, w których ten prefiks zamieniono odpowiednio na prefiks `num1`.

Parametry

pft	- struktura drzewiasta przechowująca odwrotności przekierowań;
num1	- wskaźnik na napis reprezentujący prefiks numerów, na które jest wykonywane przekierowanie;
num2	– wskaźnik na napis reprezentujący prefiks numerów przekierowywanych.

Oto graf wywołań dla tej funkcji:

5.8.2.7 bool onlyDigits (const char * *s*)

Sprawdza, czy napis jest numerem telefonu.

Sprawdza, czy napis reprezentuje poprawny numer telefonu tj. czy jest niepustym ciągiem, składającym się z cyfr 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Parametry

s	- wskaźnik na dowolny napis.

Zwraca

Wartość `true`, jeśli napis jest poprawnym numerem telefonu. Wartość `false`, jeśli napis nie jest poprawnym numerem telefonu.

5.8.2.8 void phfwdTrieFromToDelete (struct PhoneForwardTrieFromTo ** *pft*)

Usuwa strukturę.

Usuwa strukturę drzewiastą przechowującą przekierowania. Usuwa rekurencyjnie synów. Usuwa numer, na który jest przekierowywany numer reprezentowany przez dany wierzchołek.

Parametry

pft	- wskaźnik na wskaźnik na stukturę przechowującą przekierowania.

Oto graf wywołań dla tej funkcji:

5.8.2.9 struct PhoneForwardTrieFromTo* phfwdtrieFromToNew (void)

Tworzy nową strukturę.

Tworzy nową strukturę posiadającą jeden wierzchołek i brak przekierowania.

Zwraca

Wskaźnik na utworzoną strukturę lub NULL, gdy nie udało się zaalokować pamięci.

5.8.2.10 void phfwdTrieToFromDelete (struct PhoneForwardTrieToFrom ** *pft*)

Usuwa strukturę.

Usuwa strukturę drzewiastą przechowującą odwrotności przekierowań. Usuwa rekurencyjnie synów. Usuwa strukturę przechowującą numery, towarzyszącą każdemu wierzchołkowi.

Parametry

pft	- wskaźnik na wskaźnik na stukturę przechowującą przekierowania.

Oto graf wywołań dla tej funkcji:

5.8.2.11 struct PhoneForwardTrieToFrom* phfwdtrieToFromNew (void)

Tworzy nową strukturę.

Tworzy nową strukturę posiadającą jeden wierzchołek ze strukturą niezawierającą numerów telefonów.

Zwraca

Wskaźnik na utworzoną strukturę lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.8.2.12 bool phnumAdd (struct PhoneNumbers * *pnum*, char const * *num*)

Dodaje numer.

Dodaje napis reprezentujący numer telefonu do struktury przechowującej numery.

Parametry

pnum	- wskaźnik na strukturę przechowującą numery telefonów;
num	- wskaźnik na napis reprezentujący numer telefonu.

Zwraca

Wartość `true`, jeśli numer został dodany. Wartość `false`, jeśli nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.8.2.13 bool phnumAddLexic (struct PhoneNumbers * *pnum*, char * *num*)

Dodaje numer.

Dodaje napis reprezentujący numer do tablicy znajdującej się w strukturze, która przechowuje numery telefonów, zachowując porządek leksykograficzny w tablicy.

Parametry

pnum	- wskaźnik na strukturę przechowującą numery telefonów;
num	- wskaźnik na napis.

Zwraca

Wartość `true`, jeśli numer został dodany. Wartość `false`, jeśli nie udało się zaalokować pamięci.

5.8.2.14 struct PhoneNumbers* phnumNew (void)

Tworzy nową strukturę.

Tworzy nową strukturę niezawierającą numerów telefonów.

Zwraca

Wskaźnik na utworzoną strukturę lub NULL, gdy nie udało się zaalokować pamięci.

5.8.2.15 char* strdup (const char * *p*)

Tworzy kopię napisu.

Parametry

p	- wskaźnik na napis;

Zwraca

- wskaźnik na kopię napisu.

5.8.2.16 void traverse (struct **PhoneForwardTrieFromTo** * *tmp1,* struct **PhoneForwardTrieToFrom** * *tmp2,* struct **Num** * *n*)

Usuwa przekierowania.

Szuka i usuwa wszystkie przekierowania w drzewie odwrotności przekierowań `tmp2`, które są zawarte w drzewie przekierowań `tmp1`.

Parametry

tmp1	- wskaźnik na drzewo przekierowań;
tmp2	- wskaźnik na drzewo odwrotności przekierowań;
n	- wskaźnik na strukturę przechowującą numer telefonu.

Oto graf wywołań dla tej funkcji:

5.9 Dokumentacja pliku utils.h

Interfejs modułu funkcji pomocniczych dla modułu operacji przekierowywania numerów telefonów.

```
#include <stdbool.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "phone_forward.h"
```

Wykres zależności załączania dla utils.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Struktury danych

- struct PhoneForwardTrieFromTo

  Struktura drzewiasta do przechowywania numerów zgodnie z prefiksami oraz ich przekierowań.
- struct PhoneForwardTrieToFrom

  Struktura drzewiasta do przechowywania numerów zgodnie z prefiksami oraz przekierowań na dany numer.
- struct Num

  Struktura przechowująca numer telefonu.

Definicje

- #define NUMBER_OF_DIGITS 12

  Liczba różnych cyfr.
- #define INIT_SIZE 5

  Początkowy rozmiar tablicy w strukturach zawierających tablicę.
- #define MULTIPLIER 2

  Mnożnik użyty do dynamicznej alokacji pamięci.

Funkcje

- char * strdup (const char *p)

  Tworzy kopię napisu.
- struct PhoneForwardTrieFromTo * phfwdtrieFromToNew (void)

  Tworzy nową strukturę.
- struct PhoneNumbers * phnumNew (void)

  Tworzy nową strukturę.
- struct PhoneForwardTrieToFrom * phfwdtrieToFromNew (void)

  Tworzy nową strukturę.
- int **isDigit** (char c)
- bool onlyDigits (const char *s)

  Sprawdza, czy napis jest numerem telefonu.
- void phfwdTrieToFromDelete (struct PhoneForwardTrieToFrom **pft)

  Usuwa strukturę.
- void phfwdTrieFromToDelete (struct PhoneForwardTrieFromTo **pft)

  Usuwa strukturę.
- bool phnumAdd (struct PhoneNumbers *pnum, char const *num)

  Dodaje numer.
- int idxInPhNum (struct PhoneNumbers *pnum, const char *num)

  Szuka numeru.
- void onePhfwdRemove (struct PhoneForwardTrieToFrom *pft, const char *num1, const char *num2)

  Usuwa przekierowanie.
- struct Num * newNum ()

  Tworzy strukturę.
- void addToNum (struct Num *n, char add)

  Dodaje znak.
- void cutLast (struct Num *n)

  Usuwa znak.
- void delNum (struct Num *n)

  Usuwa strukturę.
- void traverse (struct PhoneForwardTrieFromTo *tmp1, struct PhoneForwardTrieToFrom *tmp2, struct Num *n)

  Usuwa przekierowania.
- bool phnumAddLexic (struct PhoneNumbers *pnum, char *num)

  Dodaje numer.

5.9.1 Opis szczegółowy

Interfejs modułu funkcji pomocniczych dla modułu operacji przekierowywania numerów telefonów.

Autor

Magdalena Augustyńska ma370723@mimuw.edu.pl

Data

01.06.2018

5.9.2 Dokumentacja funkcji

5.9.2.1 void addToNum (struct Num * n, char add)

Dodaje znak.

Modyfikuje napis przechowywany przez strukturę dodając znak na końcu napisu.

Parametry

n	- wskaźnik na strukturę przechowującą napis;
add	- znak, który ma być dodany.

Oto graf wywołań dla tej funkcji:

5.9.2.2 void cutLast (struct Num * n)

Usuwa znak.

Usuwa ostatni znak z napisu reprezentowangeo przez strukturę.

Parametry

n	- wskaźnik na strukturę przechowującą napis.

5.9.2.3 void delNum (struct Num * n)

Usuwa strukturę.

Usuwa strukturę przechowującą numer telefonu.

Parametry

n	- wskaźnik na strukturę przechowującą numer telefonu.

5.9.2.4 int idxInPhNum (struct PhoneNumbers * *pnum*, const char * *num*)

Szuka numeru.

Szuka indeksu, pod którym znajduje się numer telefonu w tablicy przechowywanej przez strukturę.

Parametry

pnum	- wskaźnik na strukturę przechowującą numery telefonów;
num	- wskaźnik na napis reprezentujący numer telefonu.

Zwraca

Liczba całkowita reprezentująca indeks w tablicy, pod którym znajduje się numer lub -1, gdy nie znaleziono podanego numeru.

5.9.2.5 struct Num* newNum ( )

Tworzy strukturę.

Tworzy nową strukturę reprezentującą numer telefonu.

Zwraca

Wskaźnik na strukturę reprezentującą numer lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.9.2.6 void onePhfwdRemove (struct PhoneForwardTrieToFrom * *pft*, const char * *num1*, const char * *num2*)

Usuwa przekierowanie.

Usuwa przekierowanie z drzewa odwrotności przekierowań wszystkich numerów mających prefiks `num2` na numery, w których ten prefiks zamieniono odpowiednio na prefiks `num1`.

Parametry

pft	- struktura drzewiasta przechowująca odwrotności przekierowań;
num1	- wskaźnik na napis reprezentujący prefiks numerów, na które jest wykonywane przekierowanie;
num2	– wskaźnik na napis reprezentujący prefiks numerów przekierowywanych.

Oto graf wywołań dla tej funkcji:

5.9.2.7 bool onlyDigits (const char * *s*)

Sprawdza, czy napis jest numerem telefonu.

Sprawdza, czy napis reprezentuje poprawny numer telefonu tj. czy jest niepustym ciągiem, składającym się z cyfr 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Parametry

s	- wskaźnik na dowolny napis.

Zwraca

Wartość `true`, jeśli napis jest poprawnym numerem telefonu. Wartość `false`, jeśli napis nie jest poprawnym numerem telefonu.

5.9.2.8 void phfwdTrieFromToDelete (struct PhoneForwardTrieFromTo ** *pft*)

Usuwa strukturę.

Usuwa strukturę drzewiastą przechowującą przekierowania. Usuwa rekurencyjnie synów. Usuwa numer, na który jest przekierowywany numer reprezentowany przez dany wierzchołek.

Parametry

pft	- wskaźnik na wskaźnik na stukturę przechowującą przekierowania.

Oto graf wywołań dla tej funkcji:

5.9.2.9 struct PhoneForwardTrieFromTo* phfwdtrieFromToNew (void)

Tworzy nową strukturę.

Tworzy nową strukturę posiadającą jeden wierzchołek i brak przekierowania.

Zwraca

Wskaźnik na utworzoną strukturę lub NULL, gdy nie udało się zaalokować pamięci.

5.9.2.10 void phfwdTrieToFromDelete (struct PhoneForwardTrieToFrom ** *pft*)

Usuwa strukturę.

Usuwa strukturę drzewiastą przechowującą odwrotności przekierowań. Usuwa rekurencyjnie synów. Usuwa strukturę przechowującą numery, towarzyszącą każdemu wierzchołkowi.

Parametry

pft	- wskaźnik na wskaźnik na stukturę przechowującą przekierowania.

Oto graf wywołań dla tej funkcji:

5.9.2.11 struct PhoneForwardTrieToFrom* phfwdtrieToFromNew (void)

Tworzy nową strukturę.

Tworzy nową strukturę posiadającą jeden wierzchołek ze strukturą niezawierającą numerów telefonów.

Zwraca

Wskaźnik na utworzoną strukturę lub NULL, gdy nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.9.2.12 bool phnumAdd (struct PhoneNumbers * *pnum,* char const * *num*)

Dodaje numer.

Dodaje napis reprezentujący numer telefonu do struktury przechowującej numery.

Parametry

pnum	- wskaźnik na strukturę przechowującą numery telefonów;
num	- wskaźnik na napis reprezentujący numer telefonu.

Zwraca

Wartość `true`, jeśli numer został dodany. Wartość `false`, jeśli nie udało się zaalokować pamięci.

Oto graf wywołań dla tej funkcji:

5.9.2.13 bool phnumAddLexic (struct PhoneNumbers * *pnum,* char * *num*)

Dodaje numer.

Dodaje napis reprezentujący numer do tablicy znajdującej się w strukturze, która przechowuje numery telefonów, zachowując porządek leksykograficzny w tablicy.

Parametry

pnum	- wskaźnik na strukturę przechowującą numery telefonów;
num	- wskaźnik na napis.

Zwraca

Wartość `true`, jeśli numer został dodany. Wartość `false`, jeśli nie udało się zaalokować pamięci.

5.9.2.14 struct PhoneNumbers* phnumNew (void)

Tworzy nową strukturę.

Tworzy nową strukturę niezawierającą numerów telefonów.

Zwraca

Wskaźnik na utworzoną strukturę lub NULL, gdy nie udało się zaalokować pamięci.

5.9.2.15 char* strdup (const char * *p*)

Tworzy kopię napisu.

Parametry

p	- wskaźnik na napis;

Zwraca

- wskaźnik na kopię napisu.

5.9.2.16 void traverse (struct **PhoneForwardTrieFromTo** * *tmp1,* struct **PhoneForwardTrieToFrom** * *tmp2,* struct **Num** * *n*)

Usuwa przekierowania.

Szuka i usuwa wszystkie przekierowania w drzewie odwrotności przekierowań `tmp2`, które są zawarte w drzewie przekierowań `tmp1`.

Parametry

tmp1	- wskaźnik na drzewo przekierowań;
tmp2	- wskaźnik na drzewo odwrotności przekierowań;
n	- wskaźnik na strukturę przechowującą numer telefonu.

Oto graf wywołań dla tej funkcji: