

Indywidualny projekt programistyczny (Info, I rok) 17/18

Kokpit ► Kursy ► Archiwum ► Rok 2017/2018 ► Informatyka ►
Informatyczne studia I stopnia ► I rok ► IPP.INFO.I.17/18 ► Temat 9 ►
Zadanie telefony, część 3

Zadanie telefony, część 3

W trzeciej części zadania oczekujemy poprawienia ewentualnych błędów z poprzednich części oraz zmodyfikowania programu.

Modyfikujemy definicję numeru telefonu. Teraz operujemy na 12 cyfrach. Numer telefonu jest to nadal niepusty ciąg, którego elementami są te cyfry. Jedenastą cyfrę, czyli cyfrę dziesięć, reprezentujemy jako znak `:`, czyli dwukropek, a dwunastą, czyli jedenaście – jako znak `;`, czyli średnik.

Moduł operacji na numerach telefonów

Mówimy, że numer telefonu `x` jest nietrywialny, jeśli wynik wykonania funkcji `phfwdReverse` dla numeru `x` zawiera jakiś numer różny od `x`. Do modułu `phone_forward` należy doimplementować funkcję

```
size_t phfwdNonTrivialCount(struct PhoneForward *pf, char const *set, size_t len);
```

Funkcja oblicza liczbę nietrywialnych numerów długości `len` zawierających tylko cyfry, które znajdują się w napisie `set`. Jeśli wskaźnik `pf` ma wartość `NULL`, `set` ma wartość `NULL`, `set` jest pusty, `set` nie zawiera żadnej cyfry lub parametr `len` jest równy zero, wynikiem jest zero. Obliczenia należy wykonywać modulo dwa do potęgi liczba bitów reprezentacji typu `size_t`. Napis `set` może zawierać dowolne znaki.

Interfejs tekstowy

Należy doimplementować operator

- `@`

oraz operację

- `@ numer` – wypisuje jedną linię (zakończoną znakiem przejścia no nowej linii) zawierającą liczbę dziesiętną (bez zer wiodących i innych dodatkowych znaków) będącą wynikiem działania funkcji `phfwdNonTrivialCount` dla aktualnej bazy przekierowań, gdzie jako parametr `set` podano napis reprezentujący `numer`, a wartość `len` to `max(0, |numer| - 12)`, przy czym `|numer|` oznacza liczbę cyfr numeru `numer`.

Dokumentacja

Cały kod należy udokumentować w formacie `doxygen`.

Dostarczamy

Rozwiązanie części 3 zadania powinno korzystać z własnego rozwiązania poprzednich jego części.

Wymagamy

Jako rozwiązanie części 3 zadania wymagamy:

- zachowania lub poprawienia struktury plików z poprzednich części,
- zmodyfikowania plików `phone_forward.h` i `phone_forward.c`,
- zmodyfikowania implementacji interfejsu tekstowego,
- uzupełnienia pliku konfiguracyjnego dla programu `cmake`,
- uzupełnienia dokumentacji w formacie `doxygen` tak, aby była przydatna dla programistów rozwijających program,

Funkcja `main` programu musi znajdować się w pliku o nazwie `phone_forward_main.c` w katalogu `src`.

Gotowe rozwiązanie powinno się kompilować w dwóch wersjach: release i debug, jak to opisano w pierwszej części zadania. W wyniku kompilacji powinien powstać plik wykonywalny `phone_forward`.

Oddawanie rozwiązania

Rozwiązanie należy oddawać, podobnie jak części 1 i 2, przez repozytorium git. W repozytorium mają się znaleźć wszystkie pliki niezbędne do zbudowania plików wykonywalnych i dokumentacji. *W repozytorium nie wolno umieszczać plików binarnych ani tymczasowych.* W Moodle jako rozwiązanie należy umieścić tekst zawierający identyfikator finalnej wersji rozwiązania, na przykład:

```
Finalna wersja mojego rozwiązania części 3 zadania telefony znajduje się
w repozytorium w wersji 518507a7e9ea50e099b33cb6ca3d3141bc1d6638.
```

Rozwiązanie należy zatwierdzić (`git commit`) i wysłać do repozytorium (`git push`) najpóźniej do godz. 20.00, 14 czerwca 2018 r.

Punktacja

Za w pełni poprawne rozwiązanie implementujące wszystkie wymagane funkcjonalności można zdobyć maksymalnie 20 punktów. Od tej oceny będą odejmowane punkty za poniższe uchybienia:

- Za problemy ze skompilowaniem rozwiązania można stracić wszystkie punkty.
- Za każdy test, którego implementacja nie przejdzie, traci się 1 punkt.
- Za problemy z zarządzaniem pamięcią można stracić do 6 punktów.
- Za niezgodną ze specyfikacją strukturę plików w rozwiązaniu, niezgodne ze specyfikacją nazwy plików w rozwiązaniu lub umieszczenie w repozytorium niepotrzebnych albo tymczasowych plików można stracić do 4 punktów.
- Za złą jakość kodu źródłowego lub błędy w stylu kodowania można stracić do 4 punktów.
- Za braki w dokumentacji można stracić do 2 punktów.
- Za ostrzeżenia wypisywane przez kompilator (`gcc` lub `doxygen`) można stracić do 2 punktów.

Rozwiązania należy implementować samodzielnie pod rygorem niezaliczenia przedmiotu.

⚙️ phone_forward_tests_12.c
📁 telefony_testy_3.zip

◀ cmocka_example.zip

Przejdź do...



Zadanie poprawkowe maraton filmowy ▶

ADMINISTRACJA



Administracja kursem

Jesteś zalogowany(a) jako Franciszek Budrowski (Wyloguj)
IPP.INFO.I.17/18

Podsumowanie zasad przechowywania danych

Pobierz aplikację mobilną

Moodle, wersja 3.5.7+ (Build: 20190823) | moodle@mimuw.edu.pl