

Wstęp do programowania w języku C

Grupa MSz w czwartki

Lista 3 na zajęcia 27.10.2022

Zadanie 1. (10 punktów na pierwszej pracowni, 5 punktów na drugiej)

Napisz program który wczytuje tekst ze standardowego wejścia i sprawdza czy wszystkie nawiasy są poprawnie sparowane. Wystarczy wypisać jedną z dwóch odpowiedzi. Zakładamy, że istnieją cztery rodzaje nawiasów: `()`, `[]`, `{}` i `<>`. Inne znaki mogą pojawić się w tekście, ale wystarczy je zignorować.

Można założyć, że długość tekstu wejściowego jest nie większa niż $2 \cdot 10^7$. Program powinien sobie poradzić z taką długością.

Przykłady:

- `(a)[b]<(c[d]{e})>` – poprawny;
- `(((((())) []){}))` – poprawny;
- `((((([aaa]]))))` – nie pasuje drugi `]`;
- `([{([({}])})]){}]` – nie pasuje ostatni `]`;
- `([])` – nie pasuje pierwszy `)`;
- `([` – niezamknięty `(`.

Przetestuj program wydajnościowo na dużym teście: raz bez włączonej optymalizacji i raz z optymalizacją (standardowo `gcc` z opcjami `-O3 -flto -march=native`).

Poniższy kod generuje tekst długości $2 \cdot 10^7$, który można zapisać do pliku a następnie podać jako wejście do swojego programu.

```
#include <stdio.h>
int main() {
    for (int i = 0; i < 4000*1000; i++) printf("(");
    for (int i = 0; i < 4000*1000; i++) printf("<");
    for (int i = 0; i < 4000*1000; i++) printf("a");
    for (int i = 0; i < 4000*1000; i++) printf(">");
    for (int i = 0; i < 4000*1000; i++) printf(")");
}
```

Aby zmierzyć czas działania można użyć polecenia `time` (na Linuxie):
`time ./program < test.txt`

Zadanie 2. (10 punktów)

Napisz program, który wczytuje dany tekst i znajduje w nim najdłuższy palindrom, czyli spójny podciąg który po odwróceniu jest identyczny. Wynikiem programu powinien być wypisany najdłuższy palindrom; w przypadku gdy jest kilka najdłuższych można wypisać dowolny z nich.

Tekst może się składać z dowolnych znaków i wczytujemy go do końca (EOF). Można założyć, że jego długość będzie nie większa niż 10^4 . (*Wskazówka*: Czyli złożoność kwadratowa wystarczy, ale sześcienna już nie.)

Przykład 1:

aabbaaccaab

Wynik:

baaccaab

Przykład 2:

aaaacb bccb

Wynik:

cb bc

Poniżej znajdują się dwa testy wydajnościowe. Oczywiście program powinien działać szybko dla każdego możliwego testu, nie tylko dla nich.

Test wydajnościowy 1 (generator):

```
for (int i = 0; i < 2500; i++) printf("a");  
for (int i = 0; i < 5000; i++) printf("b");  
for (int i = 0; i < 2499; i++) printf("a");  
printf("b");
```

Test wydajnościowy 2 (generator):

```
for (int i = 0; i < 4999; i++) printf("a");  
printf("bc");  
for (int i = 0; i < 4999; i++) printf("a");
```

Zadanie 3. *Przypominam, że do każdej listy w SKOSie jest jeszcze do zrobienia zadanie dla sprawdzaczki, które ma osobny termin.*