Linux i Python w Elektronicznej Sieci – ćwiczenia #06: Podstawy C i C++

Projekt „Matematyka dla Ciekawych Świata”,
Robert Ryszard Paciorek
<rrp@opcode.eu.org>

2023-07-03

Uwaga (1): Informacje teoretyczne przydatne do wykonania zadań oraz linki do prezentacji wideo znajdziesz w skrypcie wykładowym. Gorąco zachęcamy do korzystania z tych materiałów w trakcie rozwiązywania zadań.
Uwaga (2): Po skończeniu zadania prosimy pokazać rozwiązanie prowadzącemu - może mieć on jakiś wartościowy komentarz dotyczący tego rozwiązania.

Zadanie 1.0.1 –

Napisz funkcję przyjmującą dwa argumenty - liczbę i napis. Funkcja ma wypisać napis tyle razy ile wynosi podana liczba.

Zadanie 1.0.2 –

Napisz funkcję `podwojnie` przyjmującą napis i wypisującą go powtarzając każdy znak nie będący małą literą łacińską dwukrotnie. Np. dla 'Ala ma kota i PSA!!' funkcja powinna wypisać: 'AAla ma kota i PPSSAA!!'.
Dodatkowo funkcja powinna policzyć i zwrócić liczbę małych liter w oryginalnym napisie.
Zadanie rozwiąż używając C, nie C++. Dla ułatwienia przyjmij że napis składa się tylko z znaków ASCII.
Wskazówka: małe litery łacińskie w kodowaniach ASCII, UTF-8 i wielu innych stanowią ciągły zakres numerów znaków.

Zadanie 1.0.3 –

Jeżeli w zadaniu 1.0.2 obliczałeś długość napisu celem iteracji po jego elementach (np. użyłeś funkcji `strlen`) to rozwiąż to zadanie bez robienia tego (gdyż jest to zbędne i tylko spowalnia program).

Zadanie 1.0.4 –

Korzystając z rozwiązania zad. 1.0.2 napisz funkcję `podwojnie_tablica` przyjmująca tablicę napisów oraz ilość jej elementów. Funkcja powinna wypisać każdy napis z tej tablicy powtarzając każdą małą literę dwukrotnie.
Wskazówka: Możesz użyć (wywołać) funkcję ***`podwojnie`*** *z zadania 1.0.2 w treści* ***`podwojnie_tablica`****.*

Zadanie 2.0.1

Zmodyfikuj program z zadania 1.0.3 tak aby funkcja korzystała z arytmetyki wskaźnikowej, zamiast iteracji po numerze znaku w napisie (pętla ma korzystać z wskaźników i ich porównywania, zamiast operowania na numerze/indeksie znaku!).

Zadanie 3.0.1

Napisz funkcję `wypiszMape` (szablon funkcji) która wypisuje mapę dowolnych typów. Na przykład dla wywołania:

```
std::map<std::string, float> a = { {"xy", 1.3}, {"qw", 16.3} };
std::map<int, std::string> b = { {1, "a"}, {2, "b"} };
wypiszMape(a);
wypiszMape(b);
```

Program powinien wypisać:

```
qw → 16.3
xy → 1.3
1 → a
2 → b
```

Zadania dodatkowe

Zadanie 4.0.1 –

Zmodyfikuj rozwiązanie zadania 1.0.3 lub 2.0.1 tak aby poprawnie obsługiwało znaki kodowane jako UTF8. Np. dla 'Ą→la ma ∞ kota i PSA!' funkcja powinna wypisać: 'ĄĄ→→la ma ∞∞ kota i PPSSAA!!'.

Wskazówka 1: Zobacz opis kodowania UTF-8 na https://en.wikipedia.org/wiki/UTF-8, zauważ że w bajtach stanowiących kontynuację znaku pierwsze dwa bity mają wartość 10, natomiast pierwszy bajt znaku nigdy nie ma takiej wartości najstarszych bitów.

Wskazówka 2: Przy powtarzaniu znaku musisz wypisać wszystkie jego bajty i dopiero po tym powtórzyć to wypisywanie.

Zadanie 4.0.2

Napisz funkcję która dla otrzymanej (w argumentach) tablicy liczb zmiennoprzecinkowych, utworzy tablicę w której każdy element będzie dwukrotnie większy oraz zwróci sumę wszystkich elementów oryginalnej tablicy. Funkcja nie powinna modyfikować oryginalnej tablicy, a nową tablicę "zwracać" poprzez argument do zaalokowanej wcześniej tablicy w nim przekazanej. Zadanie rozwiąż używając C, nie C++.
Dla wywołania postaci:

```
#define daneLen 4

int main() {
        float a[daneLen] = {1.3, 3, 7.2, 13};
        float b[daneLen];
        float s = podwojnie_tablica(a, daneLen, b);
        for (int i=0; i<daneLen; ++i) {
                printf("%f ", b[i]);
        }
        printf("\nsuma = %f\n", s);
}
```

Program powinien wypisać:

```
2.600000 6.000000 14.400000 26.000000
suma = 24.500000
```

Zadanie 4.0.3

Zmodyfikuj rozwiązanie zadania 1.0.4 tak aby zamiast tablicy używać listy (`std::list`) i napisów C++ (`std::string`).

Zadanie 4.0.4

Korzystając z metody sort typu `std::list` napisz funkcję, która posortuje otrzymaną listę liczb całkowitych i zwróci sumę najmniejszego i największego elementu.

Zadanie 4.0.5

Napisz funkcję która wypisze liczby z zakresu od 0 do 20 nie podzielne przez wartość określoną w jej argumencie.

Zadanie 4.0.6

Napisz funkcję, która przyjmuje dwa argumenty: tablicę C (czyli wskaźnik na zerowy element) liczb naturalnych i jej długość. Funkcja ma wypisać każdy element tablicy oraz zwiększyć o jeden wartość tego elementu w oryginalnej tablicy. Napisz dwa warianty tej funkcji:

a) operujący arytmetyką wskaźnikową
b) operujący tablicami (czyli używający operatora dostępu do elementu tablicy: `[]`)

Zadanie 4.0.7

Napisz funkcję `zlicz` która dla podanej listy napisów policzy powtórzenia jej elementów. Na przykład dla wywołania: `std::list<std::string> l = {"AX", "B", "AX"}; zlicz(l);` program powinien wypisać:

```
AX wystepuje 2 razy
B wystepuje 1 razy
```

Wskazówka: Użyj mapy, w której element będzie stanowił klucz, a krotność jego wystąpień wartość.