

ZADANIA Z JĘZYKA C DLA GRUP 2. I 5.

Zestaw VII - styczeń 2026

20. **Porządkowanie alfabetyczne.** Proszę napisać program, który wczytuje kolejne wiersze tekstu wprowadzanego ze standardowego wejścia, aż do napotkania znacznika końca pliku (przy wprowadzaniu z klawiatury: `Ctrl-D`), zapamiętuje je w tablicy `char *Tekst[N_MAX]`; gdzie `N_MAX` to pewna ustalona maksymalna liczba wierszy, a następnie porządkuje wiersze w kolejności alfabetycznej i wyprowadza wynik na standardowe wyjście. Zadanie można stosunkowo łatwo wykonać korzystając z funkcji `qsort(...)` oraz `strcmp(...)` zdefiniowanych w plikach nagłówkowych `<stdlib.h>` i `<string.h>`. Do wczytywania kolejnych wierszy doskonale nadaje się funkcja standardowa `fgets(...)`; warto zadbać o to, aby puste wiersze, jak również wiersze zaczynające się od ustalonego znaku oznaczającego komentarz (np. znaku `'#'`), były pomijane na etapie wczytywania.
21. **Porządkowanie tabeli liczbowej.** Podobnie jak wyżej, uporządkować wiersze zawierające ciągi liczb rzeczywistych (odzielonych spacją lub przecinkiem) według wybranej (tj. pierwszej, drugiej, itp.) liczby w każdym wierszu, w porządku rosnącym lub malejącym (a mówiąc precyzyjnie: mienalejącym lub nierosnącym). Program powinien interpretować przynajmniej dwa *parametry opcjonalne* wiersza poleceń [patrz: `zestaw3i.pdf`], mianowicie: `-cN`, gdzie `N` jest numerem kolumny do przetwarzania (domyślnie będzie to *pierwsza* kolumna), oraz `-D` dla wyboru porządku malejącego (domyślnie porządkujemy rosnąco). Do wyłuskania `N`-tej liczby w wczytanego wiersza przyda się funkcja biblioteczna `strtok`. Jest także istotne, aby do porównywania liczb (można przyjąć, że będą to liczby zmiennopozycyjne podwójnej precyzji) użyć odpowiednich operatorów (`<`, `>`, itp.) po wcześniejszym przetworzeniu napisu funkcją `atof`, nie zaś funkcji dedykowanych do porównywania napisów (jak w poprzednim zadaniu).
22. **Lista wartości funkcji.** Osoby, które napisały *Kalkulator* (Zadanie 17. z Zestawu V) proszę o wykorzystanie zastosowanych tam rozwiązań w celu stworzenia programu, który wypisuje na standardowe wyjście listę wartości funkcji matematycznej wprowadzanej z klawiatury w notacji odwrotnej polskiej. Dla przykładu, $f(x) = (x - 2)^2/2$ można wprowadzić jako

`x 2 - 2 ** 2 /`

lub równoważnie

`1 2 / x 2 - 2 ** *`

Taki napis jest zapamiętywany jako tablica znakowa i przekazywany (jako argument w zmiennej globalnej) funkcji obliczającej $f(x)$. Przy każdym wywołaniu tej

funkcji (dla pewnej wartości zmiennej niezależnej x) napis jest przetwarzany podobnie jak dane wejściowe *kalkulatora*, z tym, że w przypadku napotkania znaku "x" aktualna wartość zmiennej niezależnej odkładana jest na stos (jest to zatem odpowiednik wprowadzenia liczby do kalkulatora), a po zakończeniu przetwarzania napisu funkcja zdejmuje ostatnią liczbę ze stosu i zwraca jako wynik (odpowiednik operatora "=" w kalkulatorze). Liczne fragmenty kodu kalkulatora, np. odpowiadające za operatory dwuargumentowe, będzie można wykorzystać bez zmian. Osobno (np. przed podaniem wzoru funkcji) należy wprowadzić wartość początkową (x_{start}) oraz końcową (x_{end}) zmiennej niezależnej, jak również krok Δx .

23. **Metoda bisekcji.** Proszę napisać program znajdujący miejsce zerowe funkcji metodą bisekcji (czyli *równego podziału*), o podobnej strukturze do programu z Zadania 21 (wzór funkcji wprowadzamy w notacji odwrotnej polskiej). Informacje o metodzie bisekcji wyszukać samodzielnie. Mile widziana inwencja, np. użycie innej niż bisekcja prostej metody szukania miejsca zerowego (złoty podział, *regula falsi*, itp.) Proszę zadbać o możliwie wygodne wprowadzanie parametrów niezbędnych do przeprowadzenia obliczeń (dla bisekcji będą nimi granice przedziału oraz postulowana dokładność wyniku); program powinien także wypisywać podstawowe informacje o każdym kroku algorytmu.