

Zajęcia 8

Temat: Funkcje 1

Czas trwania: 2x45 min

Cel zajęć:

projektuje i programuje proste problemy z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, tablice, łańcuchy znaków, algorytm Euklidesa, funkcje, NWD, testuje poprawność programów dla różnych danych, posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

Efekty:

- umie napisać program z wykorzystaniem instrukcji warunkowej i iteracyjnej,
- zna algorytm Euklidesa,
- umie pisać własne funkcje,
- zna zasadę podzielności, NWD,

Formy i metody pracy: praca samodzielna, omówienie, wykład

Zadania do wykonania na zajęciach	Treści programowe
1. Suma ułamków	M.2, P.2.16, A.3.1
2. Liczby super-B-pierwsze	M.2, P.2.16, A.3.1
3. Trzy liczby rosnąco	M.2, P.2.16,

Materiały do zajęć:

<https://www.main2.edu.pl/main2/courses/show/6/21/>

Zadania do wykonania w domu:

Odwróć i dodaj (przygotowana paczka do SIO2)

Liczby zaprzyjaźnione (przygotowana paczka do SIO2)

Doskonałość (przygotowana paczka do SIO2)

ZADANIA I ROZWIĄZANIA

Zadanie 1. Suma ułamków

Limit pamięci: 64MB

Napisz program, który czyta dwa ułamki zwykłe, a następnie wypisuje ułamek będący ich sumą.

Wejście

Dwa ułamki zapisane w postaci czterech liczb oddzielonych pojedynczym odstępem (odpowiednio licznik i mianownik pierwszej oraz licznik i mianownik drugiej liczby). Liczniki i mianowniki są liczbami naturalnymi nieprzekraczającymi miliarda.

Wyjście

Wypisywany ułamek (suma) w postaci licznika i mianownika przedzielonego znakiem /. Ułamek powinien być zapisany za pomocą liczb względnie pierwszych.

Przykład

Wejście 2 3 5 7	Wejście 1 4 7 4
Wyjście 29/21	Wyjście 2/1

Rozwiązanie

Rozwiązanie zadania wymaga omówienia podstawowych informacji związanych z podzielnością, NWD, algorytmem Euklidesa oraz z funkcjami, parametrami funkcji i zasięgiem zmiennych.

W zadaniu korzystamy z podstawowych umiejętności związanych z obliczaniem sumy ułamków. Dla otrzymanych licznika i mianownika należy policzyć NWD (by skrócić ułamek).

W tym celu napiszemy własną funkcję NWD korzystając z algorytmu Euklidesa: dopóki dwie liczby nie są sobie równe, od większej odejmij mniejszą; uzyskana końcowa wartość to NWD.

```
funkcja nwd (a, b)
    dopóki a ≠ b
        jeżeli a > b
            a ← a - b
        w przeciwnym wypadku
            b ← b - a
    zwróć a
```

Zadanie dla uczniów: Ile operacji musi wykonać podana funkcja w skrajnym (pesymistycznym) wypadku? Podaj taki przypadek.

Spostrzeżenie: przy dużej różnicy wartości między a i b wielokrotne odejmowanie liczby mniejszej można zastąpić operacją `mod`.

```

funkcja nwd (a, b)
    dopóki b ≠ 0
        r ← a mod b
        a ← b
        b ← r
    zwróć a

```

Nasz program wygląda więc następująco:

```

wczytaj l1, m1, l2, m2
l ← l1·m2+l2·m1
m ← m2·m1
wypisz l/nwd(l,m), '/', m/nwd(l,m)

```

Zadanie 2. Liczby super-B-pierwsze

Limit pamięci: 64MB. Źródło: cke.gov.pl

Liczba „super-B-pierwsza” to taka liczba naturalna, która spełnia następujące warunki:

- jest liczbą pierwszą,
- suma cyfr tej liczby jest liczbą pierwszą,
- suma cyfr w jej zapisie binarnym jest liczbą pierwszą.

Oblicz, ile jest liczb „super-B-pierwszych” w przedziale [a, b].

Wejście

Pierwszy wiersz danych zawiera dwie liczby całkowite a i b ($2 \leq a < b \leq 1\,000\,000$).

Wyjście

Program powinien wypisać ilość liczb „super-B-pierwszych” w zadanym przedziale.

Przykład

Wejście 2 100	Wejście 8
------------------	--------------

Rozwiązanie

Rozwiązanie zadania wymaga omówienia podstawowych informacji związanych z testowaniem pierwszości liczb oraz przypomnienie związane z systemami liczbowymi oraz z funkcjami i parametrami funkcji.

Napiszmy funkcję sprawdzającą pierwszość liczby x (brak dzielnika mniejszego lub równego \sqrt{x} poza 1). Pamiętajmy, by funkcję matematyczną pierwiastek(n) zastąpić w poniższym zapisie podnosząc strony nierówności do kwadratu (rozwiązanie wzorcowe w C++).

```

funkcja pierwsza (x)
    dla i=2 do  $\sqrt{x}$  wykonaj
        jeżeli x mod i = 0
            zwróć FAŁSZ
    zwróć PRAWDA

```

Potrzebujemy również funkcji zliczających sumę cyfr w zapisie dziesiętnym oraz binarnym. Zauważmy, że w każdym wypadku wykonujemy te same operacje zmieniając jedynie dzielnik, którym jest podstawa systemu (10 lub 2). Możemy zatem napisać jedną funkcję dla obu tych operacji:

```
funkcja suma_cyfr (x, podstawa)
    suma ← 0
    dopóki x ≠ 0
        suma ← suma + x mod podstawa
        x ← x div podstawa
    zwróć suma
```

Możemy teraz sprawdzić kolejne liczby w przedziale i wypisać wynik na wyjście:

```
wczytaj a, b
ile ← 0
dla k=a do b wykonaj
    jeżeli  pierwsza( suma_cyfr(k,10) )
        i pierwsza( suma_cyfr(k,2) )
        i pierwsza (k)
        ile ← ile + 1
wypisz ile
```

Zadanie dla uczniów: Jakie znaczenie dla liczby wykonanych operacji w zadaniu ma kolejność wywoływanych funkcji w instrukcji jeżeli?

Zadanie 3. Trzy liczby rosnąco

Limit pamięci: 64MB.

Napisz program, który czyta trzy liczby całkowite, a następnie wypisuje je w kolejności niemalejącej.

Wejście

Dane wejściowe zawierają trzy liczby oddzielone pojedynczym odstępem. Dane są dodatnimi liczbami całkowitymi nieprzekraczającymi miliona.

Wyjście

Trzy liczby.

Przykład

Wejście 7 5 3	Wejście 3 5 7
------------------	------------------

Rozwiązanie

Rozwiązanie zadania wymaga omówienia podstawowych informacji związanych z funkcjami i parametrami funkcji.

Napiszmy procedurę zamieniającą ze sobą wartościami dwie liczby, jeżeli druga liczba jest mniejsza od pierwszej (w C++ użyjemy dla funkcji typu void, parametry prześlemy przez referencję):

```
procedura zamień (&a, &b)
    jeżeli b < a
        pom ← a
        a ← b
        b ← pom
```

Teraz dla naszych trzech liczb dokonamy trzech prostych zamian.

```
wczytaj, x, y, z
zamień (x, y)
zamień (y, z)
zamień (x, y)
```

Zadanie dla uczniów: Jak moglibyśmy wykorzystać powyższą funkcję dla ciągu liczb?