

# Zadania programistyczne z Asemblera

17 listopada 2023

## 1 Operacje arytmetyczno-logiczne

- 1.1 Napisać program do wyznaczania w rejestrze EDX sumy kwadratów wartości umieszczonych w rejestrach EBX i ECX:

$$EDX \leftarrow EBX^2 + ECX^2$$

- 1.2 Napisać program do sprawdzania, czy wartość umieszczona w rejestrze ESI jest podzielnikiem wartości umieszczonej w rejestrze EDI. Wyniki — 0, jeśli jest podzielnikiem lub wartość niezerowa w przeciwnym przypadku — ma być w EAX.

- 1.3 Napisać program do wyznaczania w rejestrze EAX sumy cyfr w reprezentacji dziesiętnej wartości umieszczonej w rejestrze ESI, zakładając, że wartość ta (w reprezentacji dziesiętnej) jest co najwyżej 2-cyfrowa.

- 1.4 Napisać program do wyznaczania w rejestrze EAX sumy kwadratów cyfr w reprezentacji dziesiętnej wartości umieszczonej w rejestrze EDI, zakładając, że wartość ta (w reprezentacji dziesiętnej) jest co najwyżej 2-cyfrowa.

- 1.5 Napisać program realizujący następujące działanie:

$$ECX \leftarrow \frac{EAX^2 + EBX}{EDX}$$

- 1.6 Napisać program realizujący następujące działanie:

$$ECX \leftarrow \frac{EAX \cdot ECX - EDX^2}{-EBX}$$

## 2 Konstrukcje warunkowe

- 2.1 Zrealizować następujące operacje arytmetyczne:

(a)  $EBX \leftarrow |EDX|$

(b)  $EBX \leftarrow |ECX - EDX|$

- 2.2 Zrealizować następujące operacje:

(a)  $ECX \leftarrow \max(EAX, EBX)$

(b)  $EAX \leftarrow \max(EBX, ECX, EDX)$

- 2.3 Zrealizować następujące operacje:

(a)  $ECX \leftarrow \min(EAX, EBX)$

(b)  $EAX \leftarrow \min(EBX, ECX, EDX)$

2.4 Zrealizować następującą operację arytmetyczną:

$$EAX \leftarrow \begin{cases} -1 & , \text{ gdy } EBX < 0 \\ 0 & , \text{ gdy } EBX = 0 \\ +1 & , \text{ gdy } EBX > 0 \end{cases}$$

2.5 Zrealizować następujące operacje arytmetyczne:

- (a)  $EBX \leftarrow \lceil ESI/EDI \rceil$ , czyli iloraz wyrównany w górę do najbliższej wartości całkowitej (☛ sprawdzić resztę z dzielenia),
- (b)  $ECX \leftarrow EDI/ESI$  zaokrąglone do wartości całkowitej (☛ porównać resztę z dzielenia z połową wartości dzielnika)

2.6 Napisać program do sprawdzania, czy wartości umieszczone w rejestrach EBX, ECX i EDX mogą być długościami boków trójkąta. Wynik (wartość 1, oznaczająca TAK lub 0, oznaczająca NIE) ma zostać umieszczony w rejestrze AL.

2.7 Napisać program do sprawdzania, czy wartości umieszczone w rejestrach EBX, ECX i EDX mogą być długościami boków trójkąta prostokątnego. Wynik (wartość 1, oznaczająca TAK lub 0, oznaczająca NIE) ma zostać umieszczony w rejestrze AL.

### 3 Pętle

3.1 Zrealizować następującą operację arytmetyczną:

- (a)  $EDX \leftarrow EBX^{ECX}$  (☛ wykonać wielokrotne mnożenie),
- (b)  $EDX \leftarrow EBX!$ , czyli umieścić w EDX wartość silnia<sup>1</sup> z EBX (☛ jak wyżej).

3.2 Zrealizować następującą operację arytmetyczną:

- (a)  $EBX \leftarrow \lceil \sqrt{ECX} \rceil$ , czyli umieścić w EBX najmniejszą wartość całkowitą, której kwadrat jest większy lub równy zawartości rejestru ECX (☛ sprawdzać kolejne liczby począwszy od 1, podnosząc je do kwadratu i porównując z zawartością rejestru ECX)
- (b)  $EBX \leftarrow \lfloor \sqrt{ECX} \rfloor$ , czyli umieścić w EBX największą wartość całkowitą, której kwadrat jest mniejszy lub równy zawartości rejestru ECX (☛ jak wyżej)

3.3 Zrealizować następującą operację arytmetyczną:

$ECX \leftarrow \binom{ESI}{EDI}$ , czyli umieścić w ECX wartość współczynnika dwumiennego Newtona<sup>2</sup> (☛ w celu zmniejszenia ryzyka przekroczenia zakresu można spróbować zrealizować operacje mnożenia i dzielenia naprzemiennie, czyli  $n \cdot (n-1) \div 2 \cdot (n-2) \div 3 \cdot (n-3) \cdot \dots$ )

3.4 Napisać program do wyznaczania największego wspólnego dzielnika liczb znajdujących się w rejestrach EAX i EBX. Wynik działania ma zostać umieszczony w rejestrze EAX.

3.5 Napisać program do wyznaczania  $n$ -tego wyrazu ciągu Fibonacciego. Wartość  $n$  znajduje się w rejestrze ECX, a wynik ma się znaleźć w rejestrze EAX.

3.6 Napisać program do wyznaczania  $n$ -tego wyrazu ciągu geometrycznego. Wartość  $n$  znajduje się w rejestrze ECX, pierwszy wyraz w rejestrze EBX, iloraz w rejestrze EDX, a wynik ma się znaleźć w rejestrze EAX.

---

<sup>1</sup>  $n! = 1 \cdot 2 \cdot \dots \cdot n$

<sup>2</sup>  $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!} = \frac{n \cdot (n-1) \cdot \dots \cdot (k+1)}{1 \cdot 2 \cdot \dots \cdot (n-k)}$

- 3.7 Napisać program do wyznaczania sumy  $n$  początkowych wyrazów ciągu Fibonacciego. Wartość  $n$  znajduje się w rejestrze ECX, a wynik ma się znaleźć w rejestrze EAX.
- 3.8 Napisać program do wyznaczania sumy  $n$  początkowych wyrazów ciągu geometrycznego. Wartość  $n$  znajduje się w rejestrze ECX, pierwszy wyraz w rejestrze EBX, iloraz w rejestrze EDX, a wynik ma się znaleźć w rejestrze EAX.
- 3.9 Napisać program do wyznaczania liczby podzielników właściwych<sup>3</sup> wartości całkowitej  $n$ . Wartość  $n$  znajduje się w rejestrze EBX, a wynik ma się znaleźć w rejestrze ECX.
- 3.10 Napisać program do wyznaczania sumy podzielników właściwych wartości całkowitej  $n$ . Wartość  $n$  znajduje się w rejestrze EBX, a wynik ma się znaleźć w rejestrze ECX.
- 3.11 Napisać program do wyznaczania w rejestrze EBX sumy cyfr w reprezentacji dziesiętnej wartości umieszczonej w rejestrze ESI (➡ zadanie 1.3/str. 1).
- 3.12 Napisać program do wyznaczania w rejestrze EBX sumy kwadratów cyfr w reprezentacji dziesiętnej wartości umieszczonej w rejestrze EDI (➡ zadanie 1.4/str. 1).

## 4 Tablice

- 4.1 Napisać program do wyznaczania w rejestrze EAX średniej arytmetycznej wartości elementów tablicy liczb całkowitych
  - (a) ignorując część ułamkową,
  - (b) zaokrąglając do najbliższej liczby całkowitej (➡ zadanie 2.5b/str. 2).
- 4.2 Napisać program do wyznaczania w rejestrze EAX liczby elementów o wartości większej od 0 w tablicy liczb całkowitych.
- 4.3 Napisać program do wyznaczania w rejestrze EAX sumy elementów o wartości większej od 0 w tablicy liczb całkowitych.
- 4.4 Napisać program do wyznaczania w rejestrze EAX średniej arytmetycznej elementów o wartości większej od 0 w tablicy liczb całkowitych.
- 4.5 Napisać program do znajdowania wartości maksymalnej/minimalnej w tablicy liczb całkowitych ( $EAX \leftarrow$  wartość,  $EBX \leftarrow$  adres).
- 4.6 Napisać program do wypełniania tablicy liczb całkowitych kwadratem indeksu.
- 4.7 Napisać program do wypełniania tablicy liczb całkowitych wartościami kolejnych wyrazów ciągu Fibonacciego<sup>4</sup>. ☛ Każdy kolejny wyraz poza dwoma pierwszymi można uzyskać przez zsumowanie dwóch wyrazów bezpośrednio go poprzedzających.
- 4.8 Napisać program do sprawdzania, czy tablica liczb całkowitych jest posortowana rosnąco/malejąco. Wynik — wartość 1, oznaczająca TAK lub 0, oznaczająca NIE — ma zostać umieszczony w rejestrze AL.
- 4.9 Napisać program do sprawdzania, czy liczba całkowita nieujemna umieszczona w rejestrze ECX jest wesoła w dziesiętnym systemie pozycyjnym<sup>5</sup>. Wynik — wartość 1, gdy liczba jest wesoła lub 0 w przeciwnym przypadku — ma być w rejestrze EAX (➡ zadanie 3.12/str. 3).  
☛ Można wykorzystać stos.

<sup>3</sup> Chodzi o podzielniki począwszy od 1, ale bez samej wartości, dla której ta suma jest wyznaczana, np. dla liczby 12 są to 1, 2, 3, 4, 6; dla liczby 15 są to: 1, 3, 5.

<sup>4</sup> Dwa pierwsze wyrazu ciągu Fibonacciego mają wartość 1, a każdy następny jest sumą dwóch poprzednich.

<sup>5</sup> Liczba jest wesoła, gdy suma kwadratów jej cyfr redukuje się do 1, np. 7, 19. ➡ Wikipedia

## 5 Napisy

5.1 Napisać program do wyznaczania w rejestrze EAX liczby małych/wielkich liter w napisie (łańcuchu znaków zakończonych wartością 0).

5.2 Napisać program do zamiany

- (a) małych liter na duże,
- (b) dużych liter na małe

w napisie (łańcuchu znaków). ☛ W celu zmiany wielkości litery można oczywiście wykorzystać operacje arytmetyczne, ale również bitowe, uwzględniając fakt, że różnica w kodach ASCII pomiędzy odpowiadającymi sobie małą i wielką literą wynosi  $32 = 2^5$ .

5.3 Napisać program do konwersji liczby całkowitej nieujemnej reprezentowanej tekstowo w dziesiętnym systemie pozycyjnym na jej postać wewnętrzną (binarną) i zapisać ją w rejestrze EAX.

5.4 Napisać program do konwersji wartości całkowitej nieujemnej zapisanej w EAX na postać tekstową w dziesiętnym systemie pozycyjnym.

## 6 Podprogramy

### 6.1 Przekazywanie parametrów typu prostego przez rejestry

6.1.1 Zaimplementować i przetestować funkcję do wyznaczania wartości bezwzględnej. Parametr i wynik przekazywane są przez rejestr EAX.

6.1.2 Zaimplementować i przetestować funkcję do wyznaczania wartości znaku argumentu (➡ 2.4/str. 2). Parametr i wynik przekazywane są przez rejestr EAX.

6.1.3 Zaimplementować i przetestować funkcję do wyznaczania wartości maksymalnej/minimalnej z dwóch argumentów, przekazanych przez rejestry EAX i EBX. Wynik przekazywany jest przez rejestr EBX.

6.1.4 Zaimplementować i przetestować funkcję do sprawdzania, czy wartości parametrów, umieszczone w rejestrach EBX, ECX i EDX, mogą być długościami boków trójkąta. Wynik (wartość 1, oznaczająca TAK lub 0, oznaczające NIE) zwracany jest przez rejestr AL.

### 6.2 Przekazywanie parametrów tablicowych przez rejestry

6.2.1 Zaimplementować i przetestować funkcję do wyznaczania średniej arytmetycznej wartości elementów tablicy liczb całkowitych 32-bitowych, przekazanej odpowiednio przez parametry za pośrednictwem rejestrów procesora.

6.2.2 Zaimplementować i przetestować procedurę do wypełniania tablicy liczb całkowitych 32-bitowych, przekazanej odpowiednio przez parametry za pośrednictwem rejestrów procesora, kolejnymi wyrazami ciągu Fibonacci'ego.

6.2.3 Zaimplementować i przetestować funkcję do wyznaczania długości pierwszego słowa w napisie, przekazanym odpowiednio przez parametr jako *łańcuch znaków zakończony zerem*, za pośrednictwem rejestru procesora. Separatorami słów są: znak spacji i tabulatora.

6.2.4 Zaimplementować i przetestować procedurę do konwersji małych liter na wielkie w napisie, przekazanym odpowiednio przez parametr jako *łańcuch znaków zakończony zerem*, za pośrednictwem rejestru procesora.

### 6.3 Przekazywanie parametrów przez stos

- 6.3.1 Napisać funkcję do wyznaczania sumy wartości elementów tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry.
- 6.3.2 Napisać funkcję do wyznaczania średniej arytmetycznej elementów tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, wykorzystując w tym celu funkcję z rozwiązania zadania 6.3.1. Wynik można podać z dokładnością do liczby całkowitej
- (a) ignorując część ułamkową
  - (b) zaokrąglając do najbliższej liczby całkowitej (wskazówka: porównać resztę z dzielenia z połową wartości dzielnika).
- 6.3.3 Napisać funkcję do wyznaczania sumy elementów większych od 0 w tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry.
- 6.3.4 Napisać funkcję do wyznaczania sumy elementów aż do wystąpienia pierwszej wartości 0 w tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry.
- 6.3.5 Napisać funkcję do wyznaczania liczby elementów większych od 0 w tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry.
- 6.3.6 Napisać funkcję do wyznaczania indeksu pozycji, na której występuje pierwsza wartość 0 w tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry.
- 6.3.7 Napisać funkcję do wyznaczania wskaźnika pozycji w tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, na której występuje pierwsza wartość 0.
- 6.3.8 Napisać funkcję do wyznaczania średniej arytmetycznej elementów większych od 0 w tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry (podpunkty jak w zadaniu 6.3.2).
- 6.3.9 Napisać funkcję do wypełniania tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, wartościami kwadratu indeksu (czyli kolejno: 0, 1, 4, 9, 16, 25, ...).
- 6.3.10 Napisać funkcję do wypełniania tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, kolejnymi potęgami wartości 2 (wartościami  $2^{\text{indeks}}$ , czyli kolejno: 1, 2, 4, 8, 16, 32, ...). Wskazówka: każdą kolejną wartość można uzyskać mnożąc wartość elementu poprzedniego przez 2.
- 6.3.11 Napisać funkcję do wypełniania tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, wartościami funkcji silnia z indeksu (czyli kolejno: 1, 1, 2, 6, 24, 120, ...). Wskazówka: każdą kolejną wartość można uzyskać mnożąc wartość element poprzedniego przez indeks elementu bieżącego.
- 6.3.12 Napisać funkcję do wypełniania tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, wartościami kolejnych wyrazów ciągu arytmetycznego. Parametrami funkcją są: dane o ciągu (wyraz początkowy oraz różnica) i identyfikator tablicy (adres początkowy oraz rozmiar). Wskazówka: każdy kolejny wyraz ciągu można uzyskać dodając różnicę do wyrazu poprzedniego.
- 6.3.13 Napisać funkcję do wypełniania tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, wartościami kolejnych wyrazów ciągu geometrycznego. Parametrami funkcją są: dane o ciągu (wyraz początkowy oraz iloraz) i identyfikator tablicy (adres początkowy oraz rozmiar). Wskazówka: każdy kolejny wyraz można uzyskać mnożąc wyraz poprzedni przez iloraz.

- 6.3.14 Napisać funkcję do wypełniania tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, wartościami kolejnych wyrazów ciągu Fibonacciego<sup>6</sup>. Wskazówka: każdy kolejny wyraz poza dwoma pierwszymi można uzyskać przez zsumowanie dwóch wyrazów bezpośrednio go poprzedzających.
- 6.3.15 Napisać funkcję do sprawdzania, czy tablica liczb całkowitych 32-bitowych/16-bitowych jest posortowana rosnąco.
- 6.3.16 Napisać funkcję do wyznaczania wartości maksymalnej (największej) w tablicy liczby całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry.
- 6.3.17 Napisać funkcję do wyznaczania liczby początkowych elementów tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, których suma przekroczy wartość przekazanego w tym celu dodatkowego parametru.
- 6.3.18 Napisać funkcję (procedurę) do zamiany wartości dwóch zmiennych całkowitych 32-bitowych/16-bitowych, wskazanych przez parametry. Uwaga: parametry muszą być wskaźnikami (adresami).
- 6.3.19 Napisać funkcję do odwracania kolejności elementów tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry. W implementacji wykorzystać procedurę z rozwiązania zadania 6.3.18.
- 6.3.20 Napisać funkcję do rotacji w prawo elementów tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry. W implementacji wykorzystać procedurę z rozwiązania zadania 6.3.18.
- 6.3.21 Napisać funkcję do wyznaczania indeksu pierwszej pozycji, na której wartości w dwóch tablicach (liczb całkowitych 32-bitowych/16-bitowych) o takim samym rozmiarze, przekazanych odpowiednio przez parametry, różnią się.
- 6.3.22 Napisać funkcję do wyznaczania liczby dzielników właściwych<sup>7</sup> wartości całkowitej 32-bitowej/16-bitowej, przekazanej jako parametr.
- 6.3.23 Napisać funkcję do wyznaczania sumy dzielników właściwych wartości całkowitej 32-bitowej/16-bitowej przekazanej jako parametr.
- 6.3.24 Napisać funkcję do sprawdzania, czy liczba całkowita 32-bitowa/16-bitowa, przekazana jako parametr, jest doskonała<sup>8</sup>. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.23.
- 6.3.25 Napisać funkcję do wyznaczania liczby pozycji w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, na których występuje liczba doskonała. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.24.
- 6.3.26 Napisać funkcję do sprawdzania, czy dwie liczby całkowite 32-bitowe/16-bitowe, przekazane jako parametry, są zaprzyjaźnione<sup>9</sup>. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.23.

---

<sup>6</sup>Dwa pierwsze wyrazu ciągu Fibonacciego mają wartość 1, a każdy następny jest sumą dwóch poprzednich.

<sup>7</sup>Chodzi o dzielniki począwszy od 1, ale bez samej wartości, dla której ta suma jest wyznaczana, np. dla liczby 12 są to 1, 2, 3, 4, 6; dla liczby 15 są to: 1, 3, 5.

<sup>8</sup>Liczba doskonała to taka, która jest równa sumie swoich dzielników właściwych, np. 28 (dzielniki: 1, 2, 4, 7, 14, a ich suma wynosi  $1 + 2 + 4 + 7 + 14 = 28$ ).

<sup>9</sup>Liczby zaprzyjaźnione to takie dwie liczby, w przypadku których suma dzielników pierwszej liczby jest równa drugiej liczbie, a suma dzielników drugiej liczby jest równa pierwszej liczbie, np. 220 i 284.

- 6.3.27 Napisać funkcję do wyznaczania wartości bezwzględnej (modułu) z przekazanej jako parametr liczby całkowitej 32-bitowej/16-bitowej.
- 6.3.28 Napisać funkcję do zastąpienia wartości w tablicy liczb całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, ich modułami. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.27.
- 6.3.29 Napisać funkcję do sprawdzania, czy wartość całkowita 32-bitowa/16-bitowa, przekazana jako parametr, jest liczbą pierwszą. Wskazówka: sprawdzić, czy w przedziale obustronnie domkniętym od 2 do połowy wartości parametru znajduje się jakiś dzielnik — znalezienie dzielnika oznacza, że parametr nie jest liczbą pierwszą.
- 6.3.30 Napisać funkcję do sprawdzania, czy dwie liczby całkowite 32-bitowe/16-bitowe, przekazane jako parametry, są liczbami bliźniaczymi<sup>10</sup>. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.29.
- 6.3.31 Napisać funkcję do wyznaczania liczby liczb pierwszych w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.29.
- 6.3.32 Napisać funkcję do wyznaczania indeksu pierwszego wystąpienia liczby pierwszej w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.29.
- 6.3.33 Napisać funkcję do wyznaczania wskaźnika na pierwszy element w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, który jest liczbą pierwszą. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.29.
- 6.3.34 Napisać funkcję, która skopiuje liczby pierwsze znalezione w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej jako pierwszy parametr, na kolejne pozycje w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej jako drugi parametr. Należy przyjąć taki sam rozmiar obu przekazanych tablic i również przekazać go jako parametr do funkcji. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.29.
- 6.3.35 Napisać funkcję do sprawdzania, czy wartość całkowita 32-bitowa/16-bitowa, przekazana jako parametr, jest liczbą kwadratową. Wskazówka: sprawdzać kolejnych wartości od 0, aż do chwili, gdy kwadrat wartości jest większy lub równy wartości parametru — w przypadku równości parametr jest liczbą kwadratową.
- 6.3.36 Napisać funkcję do wyznaczania liczby liczb kwadratowych w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.35.
- 6.3.37 Napisać funkcję do wyznaczania indeksu pierwszego wystąpienia liczby kwadratowej w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.35.
- 6.3.38 Napisać funkcję do wyznaczania wskaźnika na pierwszy element w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej odpowiednio przez parametry, który jest liczbą kwadratową. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.35.
- 6.3.39 Napisać funkcję, która skopiuje liczby kwadratowe znalezione w tablicy wartości całkowitych 32-bitowych/16-bitowych, przekazanej jako pierwszy parametr, na kolejne pozycje w tablicy

---

<sup>10</sup>Liczby bliźniacze to dwie liczby pierwsze, których różnica wynosi 2, np. 3 i 5, 5 i 7, 11 i 13, 17 i 19, itd.

wartości całkowitych 32-bitowych/16-bitowych, przekazanej jako drugi parametr. Należy przyjąć taki sam rozmiar obu przekazanych tablic i również przekazać go jako parametr do funkcji. W implementacji wykorzystać (wywołać) funkcję z rozwiązania zadania 6.3.35.

- 6.3.40 Napisać funkcję do sprawdzania, czy kod znaku, przekazany jako parametr, reprezentuje znak interpunkcji. Jako znaki interpunkcji uwzględnić: kropkę, przecinek, średnik, dwukropek i znak zapytania.
- 6.3.41 Napisać funkcję do wyznaczania liczby znaków interpunkcyjnych w łańcuchu znaków, przekazanym jako parametr. W implementacji wykorzystać funkcję z rozwiązania zadania 6.3.40.
- 6.3.42 Napisać funkcję do sprawdzania, czy kod znaku, przekazany jako parametr, reprezentuje małą literę alfabetu.
- 6.3.43 Napisać funkcję do wyznaczania liczby małych liter w łańcuchu znaków, przekazanym jako parametr. W implementacji wykorzystać funkcję z rozwiązania zadania 6.3.42. Przedstawić implementację z wykorzystaniem:
- (a) adresowania w trybie indeksowym lub bazowym (w odniesieniu do tablicy bajtów, tryby te są równoważne),
  - (b) adresowania w trybie rejestrowym pośrednim.
- 6.3.44 Napisać funkcje do konwersji małej litery alfabetu na wielką. Parametr funkcji reprezentuje kod znak do konwersji, a wartość zwrotna jest kodem znaku po konwersji. W przypadku, gdy parametr nie był kodem małej litery, jest on wartością zwrotną (bez jakiejkolwiek modyfikacji). W implementacji można wykorzystać funkcję z rozwiązania zadania 6.3.42.
- 6.3.45 Napisać funkcję do konwersji małych liter na wielkie w łańcuchu znaków, przekazanym jako parametr. W implementacji wykorzystać funkcję z rozwiązania zadania 6.3.44. Przedstawić implementację z wykorzystaniem:
- (a) adresowania w trybie indeksowym lub bazowym (w odniesieniu do tablicy bajtów, tryby te są równoważne),
  - (b) adresowania w trybie rejestrowym pośrednim.
- 6.3.46 Napisać funkcję do sprawdzania, czy kod znaku, przekazany jako parametr, reprezentuje separator słów. Jako separatory uwzględnić: spację, tabulator i znak nowej linii.
- 6.3.47 Napisać funkcję do wyznaczania długości pierwszego słowa w łańcuchu znaków, przekazanym jako parametr. W implementacji wykorzystać funkcję z rozwiązania zadania 6.3.46.
- 6.3.48 Napisać funkcję do ograniczenia zawartości łańcucha znaków, przekazanego jako parametr, do pierwszego słowa. Wskazówka: wstawić znak końca łańcucha w miejscu pierwszego separatora słów. W implementacji wykorzystać funkcję z rozwiązania zadania 6.3.46 lub 6.3.47.
- 6.3.49 Napisać funkcję do wyznaczania liczby słów w łańcuchu znaków, przekazanym jako parametr. W implementacji wykorzystać funkcję z rozwiązania zadania 6.3.46. Uwaga: w łańcuchu może być kilka separatorów obok siebie, co oznacza, że nie każdy separator oddziela dwa słowa.
- 6.3.50 Napisać funkcję do sprawdzania, czy dwa łańcuchy znaków, przekazane jako parametry, są identyczne. Wskazówka: porównywać znak po znaku oba łańcuchy, aż do wykrycia pierwszej różnicy lub zakończenia łańcucha. Jeśli w ten sposób osiągnięty zostanie jednocześnie koniec obu łańcuchów, są one identyczne.