

1. Napisz program wypisujący w kolejnych wierszach standardowego wyjścia pojedyncze słowa następującego napisu „Bardzo długi napis”.
2. Napisz program, który wczytuje ze standardowego wejścia liczbę całkowitą i wypisuje ją na standardowym wyjściu.
3. Napisz program, który wczytuje ze standardowego wejścia liczbę wymierną i wypisuje ją na standardowym wyjściu
4. Napisz program, który wczytuje ze standardowego wejścia liczbę całkowitą i wypisuje na standardowym wyjściu liczbę o jeden większą.
5. Napisz program, który wczytuje ze standardowego wejścia trzy liczby całkowite i wypisuje na standardowym wyjściu ich średnią arytmetyczną.
6. Napisz program, który wczytuje ze standardowego wejścia liczbę wymierną  $x$  i wypisuje na standardowym wyjściu wartość bezwzględną z  $x$ .
7. Napisz program, który wczytuje ze standardowego wejścia liczbę wymierną i wypisuje ją na standardowym wyjściu z dokładnością do dwóch miejsc po przecinku.
8. Napisz program, który wczytuje ze standardowego wejścia liczbę wymierną i wypisuje ją na standardowym wyjściu w notacji wykładniczej (czyli takiej, w której 0.2 to 2.0e1).
9. Napisz program, który wczytuje ze standardowego wejścia liczbę całkowitą  $n$  i wypisuje na standardowe wyjście wartość bezwzględną z  $n$ . Do rozwiązania zadania nie używaj funkcji bibliotecznych za wyjątkiem operacji wejścia/wyjścia.
10. Napisz program, który wczytuje ze standardowego wejścia dwie liczby całkowite i wypisuje na standardowym wyjściu większą z nich (w przypadku gdy podane liczby są równe, program powinien wypisać którąkolwiek z nich).
11. Napisz program, który wczytuje ze standardowego wejścia trzy liczby całkowite i wypisuje na standardowym wyjściu największą z ich wartości (pamiętaj o przypadku gdy wszystkie podane liczby lub dwie z nich są równe).
12. Napisz program, który wczytuje ze standardowego wejścia dwie liczby całkowite i wypisuje tą o większej wartości bezwzględnej.
13. Napisz program obliczający pole trójkąta na podstawie wymiarów podanych przez użytkownika. Użytkownik powinien mieć możliwość podania długości podstawy i wysokości lub wszystkich boków trójkąta.
14. Napisz program, który wczytuje ze standardowego wejścia współczynniki układu dwóch równań liniowych z dwoma niewiadomymi i wypisuje na standardowym wyjściu rozwiązanie układu równań. W przypadku nieskończonej liczby lub braku rozwiązań program powinien wypisać na standardowym wyjściu odpowiednią informację.
15. Napisz program, który wczytuje ze standardowego wejścia współczynniki równania kwadratowego z jedną niewiadomą i wypisuje na standardowym wyjściu wszystkie rozwiązania rzeczywiste tego równania lub odpowiednią informację w przypadku braku rozwiązań.

16. Napisz program, który w zależności od wyboru użytkownika wczytuje ze standardowego wejścia wymiary: kwadratu, prostokąta lub trójkąta i wypisuje na standardowym wyjściu pole figury o wczytanych wymiarach.
17. Napisz program kalkulator, który wykonuje wybraną przez użytkownika operację arytmetyczną na dwóch wczytanych liczbach. Program powinien wczytywać dane ze standardowego wejścia i wypisywać wynik na standardowym wyjściu.
18. **Pętle**
19. Napisz program wczytujący ze standardowego wejścia dwie dodatnie liczby całkowite  $n$  i  $m$ , i wypisujący w kolejnych wierszach na standardowym wyjściu wszystkie dodatnie wielokrotności  $n$  mniejsze od  $m$ .
20. Napisz program wczytujący ze standardowego wejścia dwie dodatnie liczby całkowite  $n$  i  $m$ , i wypisujący na standardowym wyjściu  $m$  pierwszych wielokrotności liczby  $n$ .
21. Napisz program wczytujący ze standardowego wejścia trzy dodatnie liczby całkowite  $n$ ,  $m$  i  $k$ , i wypisujący w kolejnych wierszach wszystkie wielokrotności  $n$  większe od  $m$  i mniejsze od  $k$ .
22. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą  $n$  i wypisuje na standardowym wyjściu liczbę  $n!$ .
23. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą  $n$  i wypisuje na standardowym wyjściu sumę kwadratów liczb od 0 do  $n$ , czyli wartość  $0^2 + 1^2 + 3^2 + \dots + n^2$ .
24. Napisz program, który wczytuje ze standardowego wejścia liczbę całkowitą  $n$  ( $n > 2$ ) i wypisuje na standardowym wyjściu iloczyn liczb parzystych z zakresu od 2 do  $n$  (czyli  $2 * 4 * \dots * n$  dla  $n$  parzystych i  $2 * 4 * \dots * (n-1)$  w przeciwnym wypadku). Napisz program, który wczytuje ze standardowego wejścia dwie liczby całkowite  $n$  i  $m$  (zakładamy, że  $n < m$ ) i wypisuje na standardowym wyjściu liczbę  $n * \dots * m$ .
25. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą  $n$  i wypisuje na standardowym wyjściu element ciągu Fibonacciego o indeksie  $n$ .
26. Napisz program, który wczytuje ze standardowego wejścia dodatnie liczby całkowite  $n$  i  $m$ , i wypisuje na standardowym wyjściu największy wspólny dzielnik tych liczb.
27. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę  $n$  i wypisuje na standardowym wyjściu  $\lfloor \sqrt{n} \rfloor$ . Program napisz bez użycia funkcji z biblioteki matematycznej.
28. Napisz program, który wczytuje ze standardowego wejścia dodatnią liczbę  $n$  i wypisuje na standardowym wyjściu sumę wszystkich liczb mniejszych od  $n$ , włącznie pierwszych z  $n$ .
29. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą  $n$  i wypisuje na standardowym wyjściu wartość  $0! + 1! + \dots + n!$ .
30. Napisz program, który wczytuje ze standardowego wejścia liczbę  $n$  i wypisuje na standardowym wyjściu wszystkie trójki pitagorejskie (tj. trójki liczb całkowitych  $a$ ,  $b$ ,  $c$  takich, że  $a^2 + b^2 = c^2$ ), składające się z liczb mniejszych od  $n$ .

## 31. Funkcje

32. Napisz program, który wczytuje ze standardowego wejścia liczbę całkowitą  $n$  i wypisuje na standardowe wyjście wartość bezwzględną  $|z|$  z  $n$ . Do rozwiązania zadania nie używaj funkcji bibliotecznych za wyjątkiem operacji wejścia/wyjścia. W programie użyj samodzielnie zaimplementowanej funkcji liczącej wartość bezwzględną.
33. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą  $n$  i wypisuje na standardowym wyjściu liczbę  $n!$ . W programie użyj samodzielnie zaimplementowanej funkcji liczącej wartość silni.
34. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą  $n$  ( $n > 2$ ) i wypisuje na standardowym wyjściu największą liczbę  $k$  taką, że  $k$  dzieli  $n$  i  $k < n$ . Algorytm wyszukiwania liczby  $k$  spełniającej powyższe warunki umieść w oddzielnej funkcji.
35. Napisz funkcję, która dostaje jako argument nieujemną liczbę całkowitą  $n$  i zwraca jako wartość liczbę  $2^n$ .
36. Napisz funkcję, która dostaje jako argument liczbę całkowitą  $n$  i zwraca jako wartość liczbę  $2^n$ .
37. Napisz funkcję, która dostaje jako argumenty nieujemne liczby całkowite  $n$  i  $m$ , z których co najmniej jedna jest różna od zera i zwraca jako wartość  $n^m$ .
38. Napisz funkcję, która dostaje jako argumenty liczby całkowite  $n$  i  $m$ , z których co najmniej jedna jest różna od zera i zwraca jako wartość  $n^m$ .
39. Napisz funkcję, która dostaje jako argumenty liczbę dodatnią  $n$  i zwraca jako wartość  $\lfloor \sqrt{n} \rfloor$ . Rozwiąż zadanie nie wykorzystując funkcji bibliotecznych.
40. Napisz funkcję, która dostaje jako argumenty liczbę całkowitą  $m$  ( $m > 1$ ) oraz nieujemną liczbę  $n$  i zwraca jako wartość  $\lfloor \sqrt[m]{n} \rfloor$ . Rozwiąż zadanie nie wykorzystując funkcji bibliotecznych.
41. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą  $n$  i wypisuje na standardowym wyjściu sumę liczb mniejszych od  $n$  i zarazem względnie pierwszych z  $n$ . Algorytm wyliczania sumy podziel na dwie funkcje.
42. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą  $n$  i wypisuje na standardowym wyjściu następującą sumę:  
 $\lfloor \sqrt{1} \rfloor + \lfloor \sqrt{2} \rfloor + \lfloor \sqrt{3} \rfloor + \dots + \lfloor \sqrt{n} \rfloor$ . Algorytm wyliczania sumy podziel na dwie funkcje.
43. Napisz funkcję, która dostaje jako argument dodatnią liczbę całkowitą  $n$  i wypisuje na standardowym wyjściu wszystkie możliwe rozkłady liczby  $n$  na sumy dwóch kwadratów dodatnich liczb całkowitych. Rozważ dwa przypadki:
- gdy „ $a^2 + b^2$ ” i „ $b^2 + a^2$ ” dla  $a \neq b$  traktujemy jako dwa równe rozkłady,
  - gdy „ $a^2 + b^2$ ” i „ $b^2 + a^2$ ” traktujemy jako ten sam rozkład i wypisujemy tylko jedno z nich.

44. Napisz funkcję, która dostaje jako argument dodatnią liczbę całkowitą  $n$  i wypisuje na standardowym wyjściu wszystkie możliwe rozkłady liczby  $n$  na sumy kwadratów dodatnich liczb całkowitych. Rozważ dwa przypadki analogiczne do tych z poprzedniego zadania
45. Napisz funkcję, która dostaje jako argumenty dodatnie liczby całkowite  $n$  i  $m$ , i wypisuje na standardowym wyjściu wszystkie możliwe rozkłady liczby  $n$  na sumę dwóch dodatnich liczb całkowitych podniesionych do potęgi  $m$ . Rozważ dwa przypadki analogiczne do tych z poprzedniego zadania.
46. Napisz funkcję, która dostaje jako argumenty dodatnie liczby całkowite  $n$  i  $m$ , i wypisuje na standardowym wyjściu wszystkie możliwe rozkłady liczby  $n$  na sumę dodatnich liczb całkowitych podniesionych do potęgi  $m$ . Rozważ dwa przypadki analogiczne do tych z poprzedniego zadania.
47. Napisz funkcję, która wczytuje ze standardowego wejścia liczbę całkowitą  $i$  i zwraca ją jako swoją wartość. Dodatkowo funkcja powinna sumować wszystkie dotychczas wczytane wartości  $i$  i przy każdym swoim wywołaniu wypisywać na standardowym wyjściu ich aktualną sumę .
48. Napisz rekurencyjną funkcję, która dla otrzymanej w argumencie nieujemnej całkowitej liczby  $n$  zwraca jako wartość  $n!$ .
49. Napisz rekurencyjną funkcję zwracającą dla otrzymanej w argumencie nieujemnej liczby całkowitej  $n$  wartość elementu o indeksie  $n$  ciągu zdefiniowanego w następujący sposób:  $a_0 = 1$ ;  $a_n = 2 * a_{n-1} + 5$  dla  $n > 0$ .
50. Napisz rekurencyjną funkcję zwracającą dla otrzymanej w argumencie nieujemnej liczby całkowitej  $n$  wartość elementu o indeksie  $n$  ciągu zdefiniowanego w następujący sposób:  $a_0 = a_1 = 1$ ;  $a_n = a_{n-1} + 2 * a_{n-2} + 3$  dla  $n > 1$
51. Napisz rekurencyjną funkcję zwracającą dla otrzymanej w argumencie nieujemnej liczby całkowitej  $n$  wartość elementu ciągu Fibonacciego o indeksie  $n$ .
52. Napisz rekurencyjną funkcję zwracającą dla otrzymanej w argumencie nieujemnej liczby całkowitej  $n$  wartość elementu o indeksie  $n$  ciągu zdefiniowanego w następujący sposób  $a_0 = a_1 = 1$ ;  $a_n = a_0 + a_1 + \dots + a_{n-1}$  dla  $n > 1$
53. Napisz funkcję rekurencyjną, która dla otrzymanej w argumencie nieujemnej liczby całkowitej  $n$  zwraca wartość elementu o indeksie  $n$  ciągu zdefiniowanego w następujący sposób  $a_0 = a_1 = 1$ ;  $a_n = a_{n-1} + n$  dla  $n$  parzystych  $a_n = a_{n-1} * n$  dla  $n$  nieparzystych.
54. Napisz funkcję rekurencyjną, która dla otrzymanej w argumencie nieujemnej liczby całkowitej  $n$  zwraca wartość elementu o indeksie  $n$  ciągu zdefiniowanego w następujący sposób  $a_0 = a_1 = a_2 = 1$ ; oraz dla  $k > 2$   $a_{3*k} = a_{3*k-1} + a_{3*k-2}$ ;  $a_{3*k+1} = 5 * a_{3*k} + 4$ ;  $a_{3*k+2} = a_{3*k+1}$ .
55. Napisz funkcję rekurencyjną, która dla otrzymanej w argumentach pary nieujemnych liczb całkowitych  $n$  i  $m$  zwraca wartość  $f(n, m)$  gdzie funkcja  $f$  jest zdefiniowana w następujący sposób:  $f(n, 0) = n$ ;  $f(0, m) = m$ ;  $f(n, m) = f(n-1, m) + f(n, m-1) + f(n-1, m-1)$  dla  $n, m > 0$ .

56. Napisz funkcję rekurencyjną, która dla otrzymanej w argumentach pary nieujemnych liczb całkowitych  $n$  i  $m$  zwraca wartość  $f(n, m)$  gdzie funkcja  $f$  jest zdefiniowana w następujący sposób:  $f(n, 0) = n$ ;  $f(n, m) = f(m, n)$ ;  $f(n, m) = n - m + f(n-1, m) + f(n, m-1)$  dla  $n > m > 0$ .
57. Napisz rekurencyjną funkcję, która dostaje jako argumenty dwie dodatnie liczby całkowite  $n$  i  $m$ , i zwraca jako wartość największy wspólny dzielnik tych liczb obliczony algorytmem Euklidesa.
58. Napisz funkcję, która dostaje jako argumenty nieujemne liczby całkowite  $n$  i  $m$ , z których co najmniej jedna jest różna od zera, i zwraca jako wartość  $n_m$ . Jeżeli drugi z argumentów nie zostanie podany, funkcja powinna zwrócić wartość  $n^2$ .
59. Napisz funkcję, która dostaje jako argumenty pięć liczb całkowitych typu `unsigned int` i zwraca jako wartość maksimum z podanych liczb. Funkcję napisz w taki sposób, żeby można było jej podać także mniejszą liczbę argumentów.
60. Napisz funkcję, która dostaje jako argumenty pięć liczb typu `unsigned int` i zwraca jako wartość sumę podanych liczb. Funkcję napisz w taki sposób, żeby liczyła sumę także dwóch, trzech i czterech argumentów.
61. Napisz funkcję, która dostaje jako argumenty pięć liczb typu `int` i zwraca jako wartość iloczyn podanych liczb. Funkcję napisz w taki sposób, żeby liczyła iloczyn także dwóch, trzech i czterech argumentów.
62. Napisz funkcję, która dla dwóch dodatnich całkowitoliczbowych argumentów  $m$  i  $n$  zwraca wartość `true` jeżeli  $n$  dzieli  $m$  oraz `false` w przeciwnym wypadku. W przypadku podania jednego argumentu funkcja powinna sprawdzać czy podana liczba jest parzysta.
63. Napisz rodzinę dwuargumentowych funkcji `pot`, z których każda jako argumenty otrzymuje liczbę  $n$  i nieujemną liczbę całkowitą  $m$  typu `unsigned int` (zakładamy, że co najmniej jeden z argumentów jest różny od zera) i zwraca jako wartość  $n^m$ . Przeciąż funkcję `pot` dla  $n$  o typach: `double`, `int`, `unsigned int`. Wynik zwrócony przez każdą z funkcji `pot` powinien być tego samego typu co  $n$ .
64. **Wskaźniki i referencje**
65. Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zwraca jako wartość mniejszą z liczb wskazywanych przez argumenty.
66. Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zwraca jako wartość wskaźnik na zmienną przechowującą mniejszą z liczb wskazywanych przez argumenty.
67. Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zamienia ze sobą wartości wskazywanych zmiennych.
68. Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zamienia ze sobą wartości wskazywanych zmiennych tylko wtedy, gdy wskazywana przez drugi argument zmienna jest mniejsza od zmiennej wskazywanej przez pierwszy argument.
69. Napisz funkcję, której argumentami są dwa wskaźniki do stałych typu `int`, zaś zwracaną wartością jest suma wartości zmiennych wskazywanych przez argumenty.

70. Napisz funkcję, której argumentami są `n` typu `int` oraz `w` wskaźnik do `int`, która przepisuje wartość `n` do zmiennej wskazywanej przez `w`.
71. Napisz funkcję otrzymującą jako argumenty referencje do dwóch zmiennych typu `int`, która zamienia ze sobą wartości zmiennych, do których referencje dostaliśmy w argumentach.
72. Napisz funkcję otrzymującą dwa argumenty referencję `a` oraz wskaźnik `b` do zmiennych typu `int`, która zamienia ze sobą wartości zmiennych, do których wskaźnik i referencję dostała w argumentach.
73. Napisz bezargumentową funkcję, która rezerwuje pamięć dla pojedynczej zmiennej typu `int` i zwraca jako wartość wskaźnik do niej.
74. Napisz bezargumentową funkcję, która rezerwuje pamięć dla pojedynczej zmiennej typu `double` i zwraca jako wartość wskaźnik do niej.
75. Napisz funkcję, która dostaje jako argument dodatnią liczbę całkowitą `n`, rezerwuje w pamięci blok `n` zmiennych typu `int` i zwraca jako wartość wskaźnik do początku zarezerwowanego bloku pamięci.
76. Napisz funkcję, która dostaje jako argument dodatnią liczbę całkowitą `n`, rezerwuje w pamięci blok `n` zmiennych typu `double` i zwraca jako wartość wskaźnik do początku zarezerwowanego bloku pamięci.
77. Napisz funkcję o dwóch argumentach: wskaźnik na funkcję o jednym argumentie typu `int` zwracającą wartość typu `double` oraz wartość typu `int`, która zwraca wartość funkcji otrzymanej w pierwszym argumentie na liczbie całkowitej podanej w drugim argumentie.
78. Napisz funkcję, która otrzymuje trzy argumenty: dwa wskaźniki na funkcje o jednym argumentie typu `int` zwracające wartość typu `int`, wartość `n` typu `unsigned int`, i zwraca `true`, jeżeli otrzymane w argumentach funkcje są równe dla wartości od 0 do `n` i `false` w przeciwnym wypadku.
79. Napisz funkcję, która dostaje dwa argumenty: wskaźnik na stałą typu `int` i wskaźnik na zmienną typu `int`, i przepisuje zawartość stałej wskazywanej przez pierwszy argument do zmiennej wskazywanej przez drugi argument.
80. Napisz funkcję, która dostaje dwa argumenty: wskaźnik na stałą typu `int` i stały wskaźnik na zmienną typu `int`. I przepisuje zawartość stałej wskazywanej przez pierwszy argument do zmiennej wskazywanej przez drugi argument.
81. **Tablice jednowymiarowe**
82. Napisz funkcję, która otrzymuje dwa argumenty: nieujemną liczbę całkowitą `n` oraz `n`-elementową tablicę `tab` elementów typu `int` i nadaje wartość zero wszystkim elementom tablicy `tab`, zapisuje do kolejnych elementów tablicy wartości równe ich indeksom (do komórki o indeksie `i` funkcja ma zapisywać wartość `i`), podwaja wartość wszystkich elementów w tablicy `tab`, do wszystkich komórek tablicy `tab` wstawia wartości bezwzględne ich pierwotnych wartości.

83. Napisz funkcję, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz  $n$ -elementową tablicę `tab` o elementach typu `const int` i zwraca jako wartość średnią arytmetyczną elementów tablicy `tab`.
84. Napisz funkcję, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz  $n$ -elementową tablicę `tab` o elementach typu `unsigned int` i zwraca jako wartość średnią geometryczną elementów tablicy `tab`.
85. Napisz funkcję, która otrzymuje jako argument liczbę całkowitą  $n$  ( $n > 3$ ) i zwraca jako wartość największą liczbę pierwszą mniejszą od  $n$  (do wyznaczenia wyniku użyj algorytmu sita Eratostenesa).
86. Napisz funkcję, która otrzymuje trzy argumenty: dodatnią liczbę całkowitą  $n$  oraz dwie  $n$ -elementowe tablice `tab1`, `tab2` o elementach typu `int` i: przepisuje zawartość tablicy `tab1` do tablicy `tab2`, przepisuje zawartość tablicy `tab1` do tablicy `tab2` w odwrotnej kolejności (czyli element `tab1[0]` ma zostać zapisany do komórki tablicy `tab2` o indeksie  $n-1$ ).
87. Napisz funkcję, która otrzymuje cztery argumenty: dodatnią liczbę całkowitą  $n$  oraz trzy  $n$ -elementowe tablice `tab1`, `tab2` i `tab3` o elementach typu `int`,  $i$ : przypisuje elementom tablicy `tab3` sumę odpowiadających im elementów tablic `tab1` i `tab2` (do komórki tablicy `tab3` o indeksie  $i$  powinna trafić suma elementów `tab1[i]` i `tab2[i]`), przypisuje elementom tablicy `tab3` większy spośród odpowiadających im elementów tablic `tab1` i `tab2` (do komórki tablicy `tab3` o indeksie  $i$  powinien trafić większy spośród elementów `tab1[i]` i `tab2[i]`), przypisuje zawartość tablicy `tab1` do tablicy `tab2`, zawartość tablicy `tab2` do tablicy `tab3` oraz zawartość tablicy `tab3` do tablicy `tab1`.
88. Napisz funkcję, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz  $n$ -elementową tablicę `tab` o elementach typu `int` i: zwraca największą wartość przechowywaną w tablicy `tab`, zwraca najmniejszą wartość przechowywaną w tablicy `tab`, zwraca indeks elementu tablicy `tab` o największej wartości, zwraca indeks elementu tablicy `tab` o najmniejszej wartości, zwraca największą spośród wartości bezwzględnych elementów przechowywanych w tablicy `tab`, zwraca indeks elementu tablicy `tab` o największej wartości bezwzględnej.
89. Napisz funkcję, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz dwie  $n$ -elementowe tablice `tab` o elementach typu `double` przechowujące  $n$ -wymiarowe wektory i zwraca jako wartość iloczyn skalarny wektorów otrzymanych w argumentach.
90. Napisz funkcję, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz  $n$ -elementową tablicę `tab` o elementach typu `int` i: odwraca kolejność elementów tablicy `tab`, przesuwając o jeden w lewo wszystkie elementy tablicy (tak, żeby wartość elementu o indeksie  $n - 1$  znalazła się w elemencie o indeksie  $n - 2$ , wartość elementu o indeksie  $n - 2$  znalazła się w elemencie o indeksie  $n - 3$ , zaś wartość elementu o indeksie  $0$  w elemencie o indeksie  $n-1$ ), przesuwając o jeden w prawo wszystkie elementy tablicy (tak, żeby wartość elementu o indeksie  $0$  znalazła się w elemencie o

indeksie 1, wartość elementu o indeksie 1 znalazła się w elemencie o indeksie 2, zaś wartość elementu o indeksie n - 1 w elemencie o indeksie 0), sortuje rosnąco elementy tablicy tab (porządkuje elementy przechowywane w tablicy w taki sposób aby ciąg tab[0], tab[1],... ,tab[n-1] był ciągiem niemalejącym), sortuje malejąco elementy tablicy tab.

91. Napisz funkcję, która otrzymuje jako argument dodatnią liczbę całkowitą n, a następnie tworzy dynamiczną n-elementową tablicę o elementach typu int i zwraca jako wartość wskaźnik do jej pierwszego elementu.
92. Napisz funkcję, która otrzymuje jako argument dodatnią liczbę całkowitą n, a następnie tworzy dynamiczną n-elementową tablicę o elementach typu double i zwraca jako wartość wskaźnik do jej pierwszego elementu.
93. Napisz funkcję, która dostaje jako argument wskaźnik do jednowymiarowej dynamicznej tablicy o elementach typu int i zwalnia pamięć zajmowaną przez przekazaną w argumencie tablicę.
94. Napisz funkcję, która dostaje jako argument wskaźnik do jednowymiarowej dynamicznej tablicy o elementach typu double i zwalnia pamięć zajmowaną przez przekazaną w argumencie tablicę.
95. Napisz funkcję, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą n oraz n-elementową tablicę tab o elementach typu double a następnie tworzy kopię tablicy tab i zwraca jako wartość wskaźnik do nowo utworzonej kopii.
96. Napisz funkcję, która otrzymuje trzy argumenty: dodatnią liczbę całkowitą n oraz dwie tablice n-elementowe o elementach typu int przechowujące współrzędne wektorów i zwraca jako wartość wskaźnik do pierwszego elementu nowo utworzonej tablicy przechowującej sumę wektorów otrzymanych w argumentach.
97. Napisz funkcję, która dostaje w argumentach dodatnią liczbę całkowitą n oraz n-elementową tablicę liczb całkowitych tab1 o elementach typu int i przepisuje do nowo utworzonej tablicy tab2 elementy tablicy tab1 o wartości różnej od zera. Rozmiar tablicy tab2 powinien być równy liczbie niezerowych elementów tablicy tab1. Jako wartość funkcja powinna zwrócić wskaźnik na pierwszy element tablicy tab2.

## 98. Napisy

99. Napisz funkcję wyczysc, która usuwa z tablicy przechowywany w niej napis (w sensie: umieszcza w niej poprawny napis o długości 0).
100. Napisz funkcję dlugosc, która jako argument otrzymuje napis i zwraca jako wartość jego długość.
101. Napisz funkcję porownaj, która jako argumenty otrzymuje dwa napisy i zwraca 1 gdy napisy są równe i 0 w przeciwnym przypadku.
102. Napisz funkcję, która jako argumenty otrzymuje dwa napisy i zwraca wartość 1, gdy pierwszy napis jest wcześniejszy w kolejności leksykograficznej i 0 w przeciwnym przypadku. Zakładamy, że oba napisy składają się ze znaków typu char,



zawierają wyłącznie małe litery alfabetu łacińskiego, a system, na którym jest kompilowany i uruchamiany program, używa standardowego kodowania ASCII.

103. Napisz funkcję `przepisz`, która otrzymuje dwie tablice znaków i przepisuje napis znajdujący się w pierwszej tablicy do drugiej tablicy. Zakładamy, że w drugiej tablicy jest wystarczająco dużo miejsca.
104. Napisz funkcję `kopiujn`, która dostaje w argumentach dwie tablice znaków `nap1`, `nap2` oraz liczbę `n` i przekopiuje `n` pierwszych znaków z napisu przechowywanego w tablicy `nap1` do tablicy `nap2`. W przypadku gdy napis w tablicy `nap1` jest krótszy niż `n` znaków, funkcja powinna po prostu przepisać cały napis. Funkcja powinna zadbać o to, żeby na końcu napisu w tablicy `nap2` znalazł się znak o kodzie 0. Zakładamy, że w docelowej tablicy jest wystarczająco dużo miejsca.
105. Napisz funkcję `sklej` otrzymującą jako argumenty trzy tablice znaków i zapisującą do trzeciej tablicy konkatencję napisów znajdujących się w dwóch pierwszych tablicach (czyli dla napisów "Ala m" i "a kota" znajdujących się w pierwszych dwóch argumentach do trzeciej tablicy powinien zostać zapisany napis "Ala ma kota"). Zakładamy, że w trzeciej tablicy jest wystarczająco dużo miejsca.
106. Napisz funkcję, która otrzymuje w argumencie napis i podmienia w nim wszystkie małe litery na duże litery. Zakładamy, że napis przechowywany jest w tablicy o elementach typu `char`, składa się wyłącznie z liter łacińskich i białych znaków, oraz że system, na którym jest kompilowany i uruchamiany program, używa standardowego kodowania ASCII.
107. Napisz funkcję `wytnij`, która dostaje jako argumenty napis oraz dwie liczby całkowite `n` i `m`, i wycina z otrzymanego napisu znaki o indeksach od `n` do `m` ( $n \leq m$ ). Otrzymany w argumencie napis może mieć dowolną liczbę znaków (w tym mniejszą od `n` lub `m`)
108. Napisz funkcję `wytnij2`, która dostaje jako argument dwa napisy `nap1` i `nap2`, i wycina z napisu `nap1` pierwsze wystąpienie w nim napisu `nap2`.
109. Napisz funkcję `wytnijzw`, która dostaje jako argument dwa napisy `nap1` i `nap2`, i wycina z napisu `nap1` wszystkie znaki występujące także w napisie `nap2`
110. Napisz funkcję `wytnijzn`, która dostaje jako argument dwa napisy `nap1` i `nap2`, i wycina z napisu `nap1` wszystkie znaki niewystępujące w napisie `nap2`.
111. Napisz funkcję `wytnijtm`, która dostaje jako argument dwa napisy `nap1` i `nap2` o równej długości i wycina z napisu `nap1` znaki równe znakom występującym na tym samym miejscu w napisie `nap2` (znak o indeksie `i` usuwamy wtedy i tylko wtedy, gdy `nap1[i]=nap2[i]`).
112. Napisz funkcję `godzina`, która dostaje w argumentach trzy liczby całkowite `godz`, `min` i `sek`, zawierające odpowiednio godziny, minuty oraz sekundy, i zwraca jako wartość napis zawierający godzinę w formacie `godz:min:sek`, w którym wartości poszczególnych pól pochodzą ze zmiennych podanych w argumentach.
113. Napisz funkcję `godzina`, która dostaje w argumentach trzy liczby całkowite `godz`, `min` i `sek`, zawierające odpowiednio godziny, minuty oraz sekundy, i zwraca jako

wartość napis zawierający godzinę w formacie godz:min:sek, w którym wartości poszczególnych pól pochodzą ze zmiennych podanych w argumentach .

114. Napisz funkcję sklej, która dostaje w argumentach trzy napisy i zwraca jako wartość napis powstały ze sklejenia napisów otrzymanych w argumentach.
115. Napisz funkcję sklej, która dostaje w argumentach trzy napisy i zwraca jako wartość napis powstały ze sklejenia napisów otrzymanych w argumentach.
116. Napisz funkcję kopiuj, która dostaje jako argumenty napis oraz tablicę znaków i przepisuje napis do otrzymanej w argumencie tablicy znaków.
117. Napisz funkcję kopiuj, która dostaje jako argumenty napis oraz wskaźnik do napisu (czyli wskaźnik do wskaźnika), tworzy nową tablicę znaków, kopiuje do niej napis zawarty w pierwszym argumencie, i przypisuje wskaźnik do nowo utworzonej tablicy do zmiennej wskazywanej przez drugi argument.
118. Napisz funkcję, która dostaje w argumencie napis i zamienia wszystkie występujące w nim małe litery na odpowiadające im duże litery.
119. Napisz funkcję, która dostaje w argumencie referencję do napisu i zamienia wszystkie występujące w nim małe litery na odpowiadające im duże litery.