

Podstawy programowania

Zadania utrwalające z programowania w C

Uwagi

- W rozwiązaniach zadań nie można używać instrukcji takich jak **goto**, **continue** czy **break** (za wyjątkiem wnętrza instrukcji **switch**), które są uważane za niezgodne z zasadami programowania strukturalnego.
- W języku C nie można przeciągać funkcji ani używać funkcji z domyślnymi wartościami argumentów. W niektórych zadaniach należy zaprojektować funkcję rekurencyjną, w wielu przypadkach ich użycie pozwala znacznie uprościć kod programu.
- Należy pamiętać, że wskaźniki i tablice są ze sobą powiązane, przy czym w języku C możemy używać jednowymiarowych tablic nie wiedząc nic o wskaźnikach ani dynamicznym zarządzaniu pamięcią, wystarczy umiejętność deklarowania tablic automatycznych. Niektóre zadania będą wymagały operowania na wskaźnikach.
- W języku C istnieją dwa rodzaje tablic dwuwymiarowych. Jeden rodzaj to tablice których elementami są tablice jednowymiarowe zaś drugi to tablice wskaźników do tablic jednowymiarowych. Możemy deklarować tablice różniące się sposobem utworzenia poszczególnych wymiarów.
- W zadaniach użyto określeń: **tablice tablic** (np.: dla typu **int ****) i **tablice wielowymiarowe** (np.: dla typów **int[][n]** lub **int[][n][m]**).
- Warto zapamiętać, że w przypadku tablic przekazanych do funkcji w jej argumentach, wewnątrz funkcji pracujemy na oryginałach tablic, a nie ich kopiach. Oznacza to, że zmiana wewnątrz funkcji wartości elementów takich tablic ma konsekwencje także na zewnątrz funkcji.
- Jest to zachowanie odmienne od argumentów funkcji o typach prostych, w przypadku których pracujemy na kopiach argumentów. Takie a nie inne zachowanie tablic wynika z faktu, że są one wskaźnikami na bloki pamięci. Skopiowany wskaźnik cały czas wskazuje na ten sam blok pamięci.
- Napisy w języku C są przechowywane w jednowymiarowych tablicach o elementach typów znakowych. Istnieją dwa podstawowe typy znakowe **char** i **wchar_t**. Standard języka C nie określa długości tych typów.
- Typ **char** jest określony jako typ wystarczający do przechowywania podstawowego zestawu znaków na danym komputerze (w praktyce **char** jest typem jednobajtowym), zaś typ **wchar_t** powinien wystarczyć do przechowywania pełnego zestawu znaków dostępnego na danym komputerze.
- W zależności od implementacji **wchar_t** pozwala na kodowanie znaków w systemie **UTF-32** (w systemach Linux) lub **UTF-16** (w systemach Windows).
- Na końcu poprawnego napisu w języku C (niezależnie od typu znaków z których się składa) znajduje się znak o kodzie **0**. Służy on do zaznaczenia końca napisu. Tablica przechowująca n -znakowy napis musi mieć co najmniej $n+1$ elementów, aby móc przechować znak kończący. Konieczność dbania o to, żeby na końcu napisu był zawsze znak kończący spoczywa na programiście.
- Operacje na napisach o elementach typu **char** można znaleźć w bibliotece **<string.h>**, zaś niektóre operacje na znakach tego typu także w bibliotece **<ctype.h>**.
- Typ **wchar_t** oraz funkcje operujące na zmiennych tego typu, odpowiadające operacjom z innych bibliotek standardowych (w tym z biblioteki **<string.h>**) znajdują się w bibliotece **<wchar.h>**. Odpowiedniki funkcji z biblioteki **<ctype.h>** operujące na typie **wchar_t** znajdują się w bibliotece **<wctype.h>**.
- Funkcje z bibliotek standardowych wymagają dbania o to, żeby używane tablice znaków były wystarczających rozmiarów.

- W języku C do operacji na plikach służą funkcje z biblioteki `<stdio.h>`, a wśród nich między innymi: **fopen()**, **fclose()**, **fwrite()**, **fread()**, **fprintf()**, **fscanf()**, **feof()** i **fseek()**.
- Otwierając plik należy określić w jakim celu plik jest otwierany (do czytania, do pisania etc.) oraz czy plik ma być otwarty w trybie binarnym (czyli jako ciąg bajtów) czy tekstowym.
- W zależności od używanego kompilatora i standardu języka, może się zdarzyć konieczność używania nieco innych funkcji lub importowania innych plików nagłówkowych.
- Do wersji **C99** do obsługi wejścia i wyjścia należy używać *tradycyjnej* funkcji **printf()** i **scanf()** we wszystkich popularnych kompilatorach.
- W standardzie **C11** i późniejszych została wprowadzona możliwość implementacji tzw. bezpiecznych (*secure*) funkcji obsługi standardowego wejścia i wyjścia (np. zapewnia mechanizmu ochrony przed przepełnieniem bufora). Nie są one jednak obligatoryjne. Takie funkcje są dostępne w kompilatorze **MSVC** (firmy Microsoft) ale nie są dostępne w kolekcji **GCC** (projekt GNU).
- W przypadku korzystania z MSVC należy korzystać z funkcji **printf_s()** i **scanf_s()** lub dodać dyrektywę: **#define _CRT_SECURE_NO_WARNINGS**.

Zadania

1. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę wymierną x i wypisuje na standardowym wyjściu $\sqrt{x}$.
2. Napisz program, który wczytuje ze standardowego wejścia trzy liczby całkowite i wypisuje na standardowym wyjściu ich średnią arytmetyczną.
3. Napisz program, który wczytuje ze standardowego wejścia liczbę wymierną i wypisuje ją na standardowym wyjściu w notacji wykładniczej (np.: 0.2 to 2.0e-1).
4. Napisz program, który wczytuje ze standardowego wejścia liczbę wymierną i wypisuje ją na standardowym wyjściu z dokładnością do dwóch miejsc po przecinku.
5. 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).
6. Napisz program obliczający pole trójkąta na podstawie wymiarów podanych przez użytkownika. Użytkownik powinien mieć możliwość podania długości podstawy i wysokości lub wszystkich boków trójkąta.
7. 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.
8. 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.
9. 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.
10. 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 .
11. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą n i wypisuje na standardowym wyjściu liczbę $n!$.
12. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą n i wypisuje na

standardowym wyjściu element ciągu Fibonacciego o indeksie n .

13. Napisz program, który wczytuje ze standardowego wejścia dodatnie liczby całkowite n i m , i wypisuje na standardowym wyjściu największy wspólny dzielnik tych liczb.
14. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę n i wypisuje na standardowym wyjściu wartość $\lfloor \sqrt{n} \rfloor$ zaokrągloną w dół do najbliższej wartości całkowitoliczbowej. Program napisz bez użycia funkcji z biblioteki matematycznej.
15. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą n i wypisuje na standardowym wyjściu wartość $0! + 1! + \dots + n!$.
16. 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. W programie użyj samodzielnie zaimplementowanej funkcji liczącej wartość bezwzględną.
17. 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.
18. 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 .
19. Napisz program, który wczytuje ze standardowego wejścia nieujemną liczbę całkowitą n i wypisuje na standardowym wyjściu następującą sumę $\lfloor \sqrt{0} \rfloor + \lfloor \sqrt{1} \rfloor + \dots + \lfloor \sqrt{n} \rfloor$. Algorytm wyliczania sumy podziel na dwie funkcje.
20. Napisz funkcję, która dostaje jako argument dodatnią liczbę całkowitą n i wypisuje na standardowym wyjściu wszystkie możliwe rozkłady liczby 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.Jeżeli zajdzie taka potrzeba, możesz w rozwiązaniu używać funkcji pomocniczych.
21. Napisz funkcję, która wczytuje ze standardowego wejścia liczbę całkowitą i zwraca ją jako swoją wartość. Dodatkowo funkcja powinna sumować wszystkie dotychczas wczytane wartości i przy każdym swoim wywołaniu wypisywać na standardowym wyjściu ich aktualną sumę.
22. Napisz rekurencyjną funkcję, która dla otrzymanej w argumencie nieujemnej całkowitej liczby n zwraca jako wartość $n!$.
23. Napisz rekurencyjną funkcję zwracającą dla otrzymanej w argumencie nieujemnej liczby całkowitej n wartość elementu ciągu Fibonacciego o indeksie n .
24. 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.
25. Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zwraca jako wartość mniejszą z liczb wskazywanych przez argumenty.
26. 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.
27. Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zamienia ze sobą wartości wskazywanych zmiennych.
28. Napisz funkcję, której argumentami są n typu `int` oraz w wskaźnik do `int`, która przepisuje wartość n do

zmiennej wskazywanej przez **w**.

29. Napisz bezargumentową funkcję, która rezerwuje pamięć dla pojedynczej zmiennej typu **int** i zwraca jako wartość wskaźnik do niej.
30. 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.
31. Napisz funkcję o dwóch argumentach:
- wskaźnik na funkcję o jednym argumencie typu **int** zwracającą wartość typu **double**,
 - wartość typu **int**,
- która zwraca wartość funkcji otrzymanej w pierwszym argumencie na liczbie całkowitej podanej w drugim argumencie.
32. 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.
33. Napisz funkcję, która dostaje dwa argumenty: wskaźnik na stałą typu **int** i stały wskaźnik na zmienną typu **int**. Funkcja przepisuje zawartość stałej wskazywanej przez pierwszy argument do zmiennej wskazywanej przez drugi argument.
34. 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.
35. 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**.
36. Napisz funkcję, która otrzymuje jako argument liczbę całkowitą **n** ($n \geq 3$) i zwraca jako wartość największą liczbę pierwszą mniejszą od **n** (do wyznaczenia wyniku użyj algorytmu sita Eratostenesa).
37. 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 $n - 1$).
38. 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.

39. 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.
40. Napisz funkcję, która dostaje jako argument wskaźnik do jednowymiarowej dynamicznej tablicy o elementach typu `int` i zwalnia pamięć zajmowaną przez przekazaną w argumencie tablicę.
41. Napisz funkcję `wyczysc()`, która usuwa z tablicy przechowywany w niej napis (w sensie: umieszcza w niej poprawny napis o długości 0). Napisz dwie wersje funkcji `wyczysc()` dla napisów składających się ze znaków typu `char` i `wchar_t`.
42. Napisz funkcję `dlugosc()`, która jako argument otrzymuje napis i zwraca jako wartość jego długość. Napisz dwie wersje funkcji `dlugosc()` dla napisów składających się ze znaków typu `char` i `wchar_t`.
43. 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**.
44. 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 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. Napisz dwie wersje funkcji: dla napisów składających się ze znaków typu `char` i `wchar_t`.
45. Napisz funkcję `sklej` otrzymującą jako argumenty trzy tablice znaków i zapisującą do trzeciej tablicy konkatenaację napisów znajdujących się w dwóch pierwszych tablicach. Zakładamy, że w trzeciej tablicy jest wystarczająco dużo miejsca. Napisz dwie wersje funkcji `sklej` dla napisów składających się ze znaków typu `char` i `wchar_t`.
46. Napisz funkcję `wytnijz()`, która dostaje jako argument dwa napisy `nap1` i `nap2`, i wycina z napisu `nap1` wszystkie znaki występujące w napisie `nap2`. Napisz dwie wersje funkcji dla napisów składających się ze znaków typu `char` i `wchar_t`.
47. Napisz funkcję, która wypisuje na standardowym wyjściu otrzymany w argumencie napis. Napisz dwie wersje funkcji: dla napisów składających się ze znaków typu `char` i `wchar_t`.
48. Napisz funkcję, która dostaje jako argument tablicę znaków i wczytuje do niej napis ze standardowego wejścia. Napisz dwie wersje funkcji: dla tablicy składających się ze znaków typu `char` i `wchar_t`.
49. Napisz funkcję `godzina`, która dostaje w argumentach trzy zmienne 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. Napisz dwie wersje funkcji `godzina`: zwracające napisy będące tablicami znaków typu `char` i typu `wchar_t`.
50. Napisz funkcję, która dostaje jako argument dodatnie liczby całkowite n i m , tworzy dynamiczną dwuwymiarową tablicę tablic elementów typu `int` o wymiarach n na m , i zwraca jako wartość wskaźnik do niej.
51. Napisz funkcję, która dostaje jako argument dodatnie liczby całkowite n i m , tworzy dynamiczną dwuwymiarową tablicę elementów typu `int` o wymiarach n na m i zwraca jako wartość wskaźnik do niej.
52. Napisz funkcję, która dostaje jako argumenty wskaźnik do dwuwymiarowej tablicy tablic elementów typu `int` oraz jej wymiary: dodatnie liczby całkowite n i m , i usuwa z pamięci otrzymaną tablicę.
53. Napisz funkcję, która dostaje jako argumenty wskaźnik do tablicy dwuwymiarowej elementów typu `int` oraz jej wymiary n i m , i usuwa z pamięci otrzymaną tablicę.
54. Napisz funkcję, która dostaje w argumentach dwuwymiarową tablicę elementów typu `int`, o pierwszym wymiarze podanym jako drugi argument funkcji oraz drugim wymiarze równym **100** i wypełnia ją zerami.

55. Napisz funkcję, która dostaje w argumentach dwuwymiarową tablicę tablic elementów typu `int` oraz jej wymiary n i m , i wypełnia ją zerami.
56. Napisz funkcję, która dostaje w argumentach tablicę trójwymiarową o elementach typu `int` o wymiarach **100x100x100**, i zwraca jako wartość sumę wartości elementów tablicy.
57. Napisz funkcję, która dostaje jako argumenty dwuwymiarową tablicę tablic o elementach typu `int` oraz jej wymiary, i zwraca jako wartość indeks wiersza o największej średniej wartości elementów. Przyjmujemy, że dwa elementy leżą w tym samym wierszu, jeżeli mają taki sam pierwszy indeks.
58. Napisz funkcję, która dostaje jako argumenty dwuwymiarową tablicę tablic o elementach typu `int` oraz jej wymiary, i zwraca największą spośród średnich wartości elementów poszczególnych wierszy. Przyjmujemy, że dwa elementy leżą w tym samym wierszu, jeżeli mają taki sam pierwszy indeks.
59. Napisz funkcję, która dostaje jako argumenty dwuwymiarową kwadratową tablicę tablic `tab` o elementach typu `int` oraz jej wymiar, i zmienia kolejność elementów w otrzymanej tablicy w następujący sposób: dla dowolnych k i j element `tab[k][j]` ma zostać zamieniony miejscami z elementem `tab[j][k]`.
60. Napisz funkcję, która otrzymuje w argumentach dwie kwadratowe tablice tablic elementów typu `int` oraz ich wspólny wymiar, i zwraca jako wartość wynik mnożenia macierzy przechowywanych w przekazanych argumentach. Wynik powinien zostać zwrócony w nowo utworzonej tablicy `tablic`.
61. Napisz funkcję, która dostaje jako argument ścieżkę dostępu do pliku, otwiera plik do tekstowego czytania i zwraca jako wartość deskryptor świeżo otwartego pliku.
62. Napisz funkcję, która dostaje jako argument deskryptor do pliku tekstowego otwartego do czytania, wypisuje zawartość pliku na standardowe wyjście i zamyka plik.
63. Napisz funkcję, która dostaje jako argument ścieżkę dostępu do pliku tekstowego i wypisuje na standardowym wyjściu zawartość pliku z pominięciem białych znaków. Napisz dwie wersje funkcji dla znaków typów `char` i `wchar_t`.
64. Napisz funkcję, która dostaje jako argumenty ścieżkę dostępu do pliku tekstowego oraz znak `c` i zwraca jako wartość liczbę wystąpień znaku `c` w podanym w argumencie pliku.
65. Napisz funkcję, która dostaje jako argumenty ścieżki dostępu do dwóch plików tekstowych i zwraca jako wartość **1**, jeżeli podane pliki mają taką samą zawartość oraz **0** w przeciwnym wypadku. Napisz dwie wersje funkcji dla znaków typu `char` i `wchar_t`.
66. Napisz funkcję, która dostaje jako argumenty ścieżki dostępu do dwóch plików tekstowych i zwraca jako wartość **1**, jeżeli podane pliki mają taką samą zawartość z dokładnością do białych znaków oraz **0** w przeciwnym wypadku.
67. Napisz funkcję, która dostaje jako argumenty deskryptory dwóch plików tekstowych i przepisuje zawartość pierwszego pliku do drugiego pliku.
68. Napisz funkcję, która dostaje jako argumenty nazwę pliku, dwuwymiarową tablicę tablic o elementach typu `int` oraz wymiary tablicy i zapisuje binarnie zawartość tablicy do podanego pliku.
69. Napisz funkcję, która dostaje jako argumenty nazwę pliku, dwuwymiarową tablicę tablic o elementach typu `int` oraz wymiary tablicy i wczytuje binarnie zawartość pliku do tablicy. Napisz funkcję tak, aby była *kompatybilna* z funkcją z poprzedniego zadania.