

Wstęp do programowania

Zestaw 7 — funkcje

Zadania realizowane na zajęciach

Zadanie 1. Napisać program, który wyznaczy pole i obwód prostokąta o długościach boków podanych przez użytkownika. W programie zdefiniować i wywołać następujące funkcje:

- o nazwie `policz_pole`, która zwraca pole prostokąta na podstawie długości boków przekazanych w argumentach.
- o nazwie `policz_obwod`, która zwraca obwód prostokąta na podstawie długości boków przekazanych w argumentach.

Zadanie 2. W programie zdefiniować funkcję o nazwie `linia` wypisującą na ekranie linię zbudowaną z podanej liczby znaków, liczbę znaków oraz typ znaku przekazać w parametrach. Następnie, wykorzystując funkcję `linia`, wyświetlić poniższe kształty, przy czym wysokość kształtu podaje użytkownik (przykłady dla $n=4$).

(a)	(b)	(c)
0000	A	%
0000	AA	%%
0000	AAA	%%%
0000	AAAA	%%%%

Zadanie 3. Napisać funkcję, która dostaje jako argumenty pięć liczb całkowitych typu `int` i zwraca jako wartość sumę podanych liczb. Funkcję napisać tak, aby można było jej podać także mniejszą liczbę argumentów. W programie zastosować funkcję w ten sposób, że użytkownik podaje liczby a, b, c, d, e , a program oblicza sumy $a+b$, $a+b+c$, $a+b+c+d$, $a+b+c+d+e$.

Zadanie 4. Stosując mechanizm przeciążania napisać funkcje, które dostają jako argument liczbę całkowitą n i zwracają jako wartość liczbę 2^n . Zilustrować działanie funkcji w programie.

Zadanie 5. Napisać rekurencyjną funkcję zwracającą dla otrzymanej w argumencie nieujemnej liczby całkowitej n wartość elementu ciągu Fibonacciego o indeksie n . Zilustrować działanie funkcji w programie.

Zadania do rozwiązania w domu

Zadanie 6. W programie zdefiniować funkcję o nazwie `cale_na_cm` przeliczającą długość w calach przekazaną w parametrze na długość w centymetrach. Wykorzystać napisaną funkcję do przeliczenia wartości podanych przez użytkownika.

Zadanie 7. Napisać program, który wczytuje ze standardowego wejścia liczbę całkowitą n i wypisuje na standardowe wyjście wartość bezwzględną z n . Do rozwiązywania zadania nie używać funkcji bibliotecznych za wyjątkiem operacji wejścia/wyjścia. W programie użyć samodzielnie zaimplementowanej funkcji liczącej wartość bezwzględną.

Zadanie 8. Napisać 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 umieścić w oddzielnej funkcji.

Zadanie 9. Napisać funkcję, która wyznacza liczbę cyfr zadanej liczby całkowitej.

Zadanie 10. Napisać funkcję, która dostaje jako argumenty pięć liczb całkowitych typu `int` i zwraca jako wartość iloczyn podanych liczb. Funkcję napisać tak, aby można było jej podać także mniejszą liczbę argumentów. W programie zastosować funkcję w ten sposób, że użytkownik podaje liczby a, b, c, d, e , a program oblicza iloczyny $a*b$, $a*b*c$, $a*b*c*d$, $a*b*c*d*e$.

Zadanie 11. Napisać funkcję, która dostaje jako argumenty pięć liczb całkowitych typu `unsigned int` i zwraca jako wartość maksimum podanych liczb. Funkcję napisać tak, aby można było jej podać także mniejszą liczbę argumentów. W programie zastosować funkcję w ten sposób, że użytkownik podaje liczby a, b, c, d, e , a program oblicza maksima $\max\{a, b\}$, $\max\{a, b, c\}$, $\max\{a, b, c, d\}$, $\max\{a, b, c, d, e\}$.

Zadanie 12. 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) a następnie zwraca jako wartość n^m . Przeciążyć funkcję `pot` dla n o typach: `double`, `int`, `unsigned int`. Wynik zwrócony przez każdą z funkcji `pot` powinien być tego samego typu co n .

Zadanie 13. Rozbuduj program z zadania 10 z zestawu 5 tak, aby można było wykonać powyższe obliczenia dla n sędziów, którzy mają do dyspozycji punkty z przedziału $[a, b]$ ($a < b$).

Zadanie 14. Napisać program wypisujący na ekranie n kolejnych wyrazów ciągu $(a_n)_{n \in \mathbb{N}}$ spełniającego równanie rekurencyjne

$$\begin{cases} a_0 = 1 \\ a_1 = 2 \\ a_{n+2} = 5a_n - 6a_{n+1}, \end{cases}$$

gdzie n jest liczbą naturalną wprowadzaną przez użytkownika.