

Pracownia programowania (2025) - Kolokwium 1 - Zestaw A12 (przykładowy)

Zasady kolokwium:

- Obowiązuje regulamin zajęć.
- Czas: 60 minut (ew. jak zostanie czasu do końca zajęć, to można zostać).
- Łącznie do zdobycia max 40 punktów. Próg zaliczenia: 25 pkt (bez innych punktów).
- **Kolokwium należy wykonać na komputerach zamontowanych na stałe w pracowniach.**
- Student przysyłając rozwiązania oświadcza, że rozwiązał je samodzielnie.
- W trakcie kolokwium nie można korzystać z żadnych materiałów pomocniczych w żadnej formie. Wszelkie kody powinny być napisane manualnie bez wspomagania się dodatkami automatycznie generującymi kod (np. Copilot, chat GPT itp.).
- Publikowanie poleceń i rozwiązań w internecie jest zabronione do czasu napisania kolokwium przez wszystkie grupy ćw.
- Kod musi się kompilować, aby był sprawdzany.
- Kod zakomentowany nie będzie sprawdzany.
- W trakcie kolokwium zostanie udostępniony przez prowadzącego pendrive. Zawartość pendrive będzie może zawierać pliki pomocnicze do poleceń. Udostępniony będzie w celu zgrania rozwiązań. Umieszczenie poleceń na pendrive powinno odbyć się w czasie kolokwium.
- Rozwiązania po czasie mogą nie być sprawdzane.
- O ile nie zaznaczono w poleceniu inaczej, każdą z funkcji należy wywołać co najmniej jeden raz (może być bardzo trywialnie).
- Należy przestrzegać nazw funkcji i kolejności argumentów w poleceniach.
- Warto zwracać uwagę na typ zwracany funkcji — jeśli metoda ma „coś” zwrócić, będzie to wskazane w poleceniu.
- W rozwiązaniach nie należy wykonywać nadmiarowych czynności, niewskazanych w poleceniu w ramach samodzielnie zdefiniowanej funkcji. Dodatkowe czynności mogą odbywać się w main. O ile w poleceniu nie zaznaczono inaczej, w rozwiązaniu można stworzyć funkcje pomocnicze.
- Po kartkach z poleceniami można pisać i traktować jako brudnopis.
- W rozwiązaniach należy uwzględniać dobre praktyki omawiane na wykładzie, o ile polecenie nie mówi coś innego.
- Format rozwiązania:
 - każde rozwiązanie w osobnym pliku z rozszerzeniem.
 - nazwa plików: zad1.c, zad2.c, zad3.c, zad4.c
 - Zadania powinny być umieszczone w archiwum .zip na udostępnionym pendrive.
 - Nazwa archiwum powinna być wg schematu
NUMERZESTAWU_NUMERALBUMU.zip
gdzie numer zestawu znajduje się na górze kartki z poleceniami. np. A23_123456.zip
- Odstępstwa od zasad są możliwe jedynie po uzyskaniu wcześniejszej zgody na tzw. alternatywne sposoby zaliczeń.

1. W folderze DebugXYZ (XYZ - losowe znaki) na pendrive znajduje się projekt z kodem w języku C. W pliku main.c w niektórych liniach są komentarze. Twoim zadaniem jest wpisanie wartości odpowiednich zmiennych po wykonaniu konkretnej linii kodu. Dopisanie nowych linii kodu czy zaburzenie struktury kodu oznacza zero punktów za polecenie.

Punktacja: 10 pkt.

2. Napisz program, który prosi użytkownika o wprowadzenie dodatniej liczby całkowitej n , a następnie oblicza i wyświetla $\lfloor \sqrt[3]{n} \rfloor$ (część całkowita/podłoga pierwiastka sześciennego). Zadanie należy rozwiązać bez tablic, wskaźników, wbudowanych funkcji matematycznych. Nie twórz samodzielnie też własnych funkcji.

Przykład: dla $n=27$ powinno być wyświetlone 3, zaś dla $n=77$ powinno być wyświetlone 4.

Punktacja: 10 pkt.

3. Napisz funkcję `maxPtr` z dwoma argumentami, które są wskaźnikami na zmienne typu `int`. Funkcja zwraca wskaźnik na zmienną o największej wartości spośród tych, na które wskazują argumenty. W przypadku gdy wskazywane wartości są równe, zwróć drugi wskaźnik. Utwórz przypadek testowy w main, aby wyświetlić najmniejszą wartość spośród trzech zmiennych.

Przykładowa sytuacja przed wywołaniem funkcji:

Nazwa zmiennej	Wartość zmiennej	Adres zmiennej
a	4.2	0x00204
b	-3.4	0x00338

Funkcja ma zwrócić dla powyższych danych 0x00204.

Punktacja: 10 pkt.

4. Napisz funkcję `shiftLeft`, której argumentami są dodatnia liczba całkowita n reprezentująca rozmiar tablicy oraz n -elementowa tablica `tab` o elementach typu `int`. Funkcja ma zwrócić wskaźnik do nowo utworzonej dynamicznej tablicy, w której wszystkie elementy zostały przesunięte o dwie pozycje w lewo w stosunku do oryginalnej tablicy, a dwa ostatnie elementy nowej tablicy są równe dwóm początkowym elementom oryginalnej tablicy. Stwórz przypadek testowy.

Na przykład, jeśli oryginalna tablica wyglądała tak: [5, 8, 3, 1, 7], to po przesunięciu o dwie pozycje w lewo będzie wyglądać tak: [3, 1, 7, 5, 8] - pierwsze dwa elementy oryginalnej tablicy trafiły na koniec, a reszta została przesunięta.

Punktacja: 10 pkt - gdy rozwiązanie w całości oparte na wskaźnikach. 7 pkt - gdy rozwiązanie bazuje na notacji tablicowej (przez nawiasy kwadratowe). W przypadku rozwiązania mieszanego, maksymalna liczba punktów może być zmienna.

