

Programowanie strukturalne (2024) - Przykładowe Kolokwium 2 - Zestaw W12

Zasady kolokwium:

- Obowiązuje regulamin zajęć.
- Czas: 90 minut (ew. jak zostanie czasu do końca zajęć, to można zostać).
- Łącznie do zdobycia max 60 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.
- Format rozwiązania:
 - każde rozwiązania w osobnym pliku z rozszerzeniem.
 - nazwa plików: zad1.c, zad2.c, zad3.c, zad4.c, zad5.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

1. W folderze DebugXYZ (XYZ - losowe znaki) 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 czy zaburzenie struktury kodu oznacza zero punktów za polecenie. W przypadku znaków, należy zapisać sam znak w apostrofach np. 'c' (wielkość znaków ma znaczenie).

Punktacja: 6 pkt.

2. Napisz funkcję, której argumentami są dwa napisy. Funkcja powinna zwrócić informację ile znaków będących małymi literami jest w krótszym napisie. Jeśli napisy są równej długości, to funkcja powinna zwrócić liczbę małych liter z pierwszego napisu. Stwórz przypadek testowy. W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia.

Punktacja: 10 pkt.

3. Napisz funkcję, której argumentem jest dwuwymiarowa kwadratowa tablica tablic (zawierająca zmienne typu `int`) oraz jej wymiar n , $n > 1$. Funkcja powinna zwrócić największy element znajdujący się na przekątnej pobocznej (suma indeksów jest równa $n - 1$). Stwórz przypadek testowy.

Punktacja: 10 pkt.

4. Stwórz strukturę `Product` o dwóch polach: `quantity` (`int`, ilość) i `price` (`float`, cena). Następnie stwórz funkcję, której argumentami jest niepusta tablica struktur `Product` oraz rozmiar tablicy. Funkcja ma zwrócić produkt (jako strukturę) o najwyższej cenie. W przypadku kilku produktów w tablicy o najwyższej cenie, to zwróć ostatni z możliwych. Stwórz przypadek testowy.

Punktacja: 12 pkt.

5. Napisz funkcję `printNegative`, która przyjmuje jako argument listę z głową o elementach typu:

```
struct element {
    int x;
    struct element * next;
};
```

Funkcja ma wyświetlić te elementy listy, które są ujemne. Stwórz przypadek testowy.

Punktacja: 22 pkt.

