Programowanie strukturalne (2024) - Przykładowe Kolokwium 2 - Zestaw W13

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 przesyłając 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 - losowowe znaki) znajduje się projekt z kodem w języku C. W pliku main.c w niektórych linijkach są komentarze. Twoim zadaniem jest wpisanie wartości odpowiednich zmiennych po wykonaniu konkretnej linii kodu. Dopisanie nowych linijek 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 `findLast`, której argumentem jest napis. Funkcja zwraca numer indeksu, na którym występuje ostatni od lewej znak cyfry. W przypadku pustego napisu lub braku znaku cyfry w napisie, funkcja powinna zwracać -1. W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia. Stwórz przypadek testowy.

Przykład: dla napisu `"abbbc39WW"` funkcja powinna zwrócić 6.

Punktacja: 10 pkt.

3. Napisz funkcję, której argumentem jest dwuwymiarowa kwadratowa tablica tablic (zawierająca zmienne typu `int`) oraz jej wymiar n. Funkcja ma zwrócić iloczyn elementów podzielnych przez 3. W przypadku braku takich elementów zwróć jeden. Stwórz przypadek testowy.

Wskazówka: Tablica tablic powinna być zadeklarowana jako podwójny wskaźnik.

Przykład. Dla poniższej tablicy zaznaczono prostokątami elementy podzielne przez 3. Funkcja zatem dla takiej tablicy powinna zwrócić $2916 = 3 \cdot (-6) \cdot 6 \cdot -3 \cdot 9$.

$$\begin{bmatrix} -2 & \boxed{3} & 4 & \boxed{-6} \\ -5 & \boxed{6} & -2 & 2 \\ 4 & \boxed{-3} & 5 & -8 \\ \boxed{9} & 7 & 0 & 4 \end{bmatrix}$$

Punktacja: 10 pkt.

4. Stwórz strukturę `Budynek` o dwóch polach `adres` (napis) oraz `numer` (`dowolny typ całkowity`). Następnie stwórz funkcję, której argumentami jest tablica struktur `Budynek` oraz rozmiar tablicy. Funkcja ma zwrócić numer budynku o najkrótszym pod kątem długości adresie. Stwórz przypadek testowy.

Przykład: Dla tablicy struktur:

indeks	adres	numer
0	`"AABBCC"`	34
1	`"XX"`	12
2	`"KLMNO"`	28

funkcja ma zwrócić 12.

Punktacja: 12 pkt.

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

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

oraz liczba całkowita `a`. Funkcja ma wyświetlić te elementy listy, które są mniejsze od od `a`. Stwórz przypadek testowy.

Wskazówka: Lista z głową ma pierwszy element, którego wartość nie jest zainicjowana.

Przykład: Dla listy zawierającej elementy 3,-4,7,12,9 oraz `a=5` funkcja ma wyświetlić: 3,-4.

Dla pustej listy funkcja ma nic nie wyświetlać.

Punktacja: 22 pkt.