Programowanie strukturalne
- Przykładowe Kolokwium 2
- Zestaw W02

Rozwiązania mają być umieszczone zgodnie ze specyfikacją:

- 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
- We wnętrzu archiwum powinny znajdować się tylko same kody w języku C, pliki powinny posiadać dokładnie nazwy (z uwzględnieniem wielkości znaków): `zad1.c`, `zad2.c`, `zad3.c`, `zad4.c`, `zad5.c`.
- Maksymalna waga archiwum 10 MB.
- Archiwum powinno być bez hasła.
- W przypadku pominięcie danego zadania, należy dodać plik o nazwie sprecyzowanej wyżej (zawartość może być pusta).

Za zachowanie specyfikacji dokładnie otrzymuję się dodatkowe 2 punkty. Zadania znacznie odbiegające od specyfikacji mogą nie być sprawdzane.

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, której argumentem jest napis. Funkcja ma z napisu usunąć wszystkie małe litery. Stwórz przypadek testowy.

Przykład: “Abecadlo” ma być zamieniony na “A”.

Punktacja: 10 pkt.

3. Napisz funkcję, której argumentem jest dwuwymiarowa kwadratowa tablica tablic (zawierająca zmienne typu `int`) oraz jej wymiar n. Funkcja ma odwrócić kolejność elementów w wierszach o nieparzystych indeksach. Stwórz przypadek testowy.

Przykład.

$$\begin{bmatrix} 2 & 3 & -3 & 1 \\ 1 & 4 & 7 & 2 \\ -3 & -6 & 11 & 3 \\ -2 & 8 & 23 & 4 \end{bmatrix} \longrightarrow \begin{bmatrix} 2 & 3 & -3 & 1 \\ 2 & 7 & 4 & 1 \\ -3 & -6 & 11 & 3 \\ 4 & 23 & 8 & -2 \end{bmatrix}$$

Punktacja: 10 pkt.

4. Stwórz strukturę `Szkola` o dwóch polach `adres` (napis) oraz numer (`dowolny typ całkowity`). Następnie stwórz funkcję, której argumentami jest niepusta tablica struktur `Szkola` oraz rozmiar tablicy. Funkcja ma wyświetlić listę szkół posortowaną wg numeru od najmniejszego do największego (uwaga: wyświetl elementy posortowane, ale nie sortuj elementów na tablicy). Stwórz przypadek testowy.

Punktacja: 12 pkt.

5. Napisz funkcję, która przyjmuje jako argument dwie listy z głową o elementach typu:

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

Funkcja zwraca 1 jeśli obie listy są mają po tyle samo elementów dodatnich oraz 0 w przeciwnym wypadku. Stwórz przypadek testowy.

Punktacja: 22 pkt.