

# Programowanie strukturalne - Kolokwium 2 - Zestaw B52

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 pominięcia 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 - 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 argumentem jest napis. Funkcja zwraca numer indeksu, na którym występuje pierwsza od lewej mała litera. W przypadku pustego napisu lub braku małych liter, funkcja powinna zwracać zero. W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia. Stwórz przypadek testowy.

*Punktacja: 10 pkt.*

3. Napisz funkcję, której argumentem jest dwuwymiarowa tablica tablic (zawierająca zmienne typu `int`) oraz jej wymiary  $n$  i  $m$ . Funkcja ma zwrócić najmniejszy element parzysty w tablicy (chodzi o wartości parzyste, a nie indeksy). W przypadku ich braku, zwróć `-1`. Stwórz przypadek testowy.

*Punktacja: 10 pkt.*

4. Stwórz strukturę `Komputer` o dwóch polach `marka` (napis) oraz `cena` (dowolny typ całkowity). Następnie stwórz funkcję, której argumentami jest niepusta tablica struktur `Komputer` oraz rozmiar tablicy. Funkcja ma zwrócić najdroższą cenę za komputer z tablicy. Stwórz przypadek testowy.

*Punktacja: 12 pkt.*

5. Napisz funkcję, która przyjmuje jako argument dwie listy bez głow o elementach typu:

```
struct elem {  
    int t;  
    struct elem * next;  
};
```

Funkcja zwraca 1 jeśli suma elementów na obu listach jest taka sama oraz 0 w przeciwnym wypadku. Stwórz przypadek testowy.

*Punktacja: 22 pkt.*