

# Programowanie strukturalne - Kolokwium 2 - Zestaw B75

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. Jeśli napis zawiera inne znaki niż cyfr, to funkcja ma zwracać zero. Jeśli napis zawiera tylko cyfry, funkcja ma zwrócić liczbę całkowitą powstałą z przepisania kolejno znaków cyfr. Załóż, że napis jest długości dokładnie 3.
3. W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia. Stwórz przypadek testowy.

Przykład. Dla napisu “345” funkcja ma zwrócić 345 (jako liczbę w typie całkowitoliczbowym).

*Punktacja: 10 pkt.*

- Napisz funkcję, której argumentem jest dwuwymiarowa tablica elementów (zawierająca zmienne typu `int`) oraz jej wymiary  $n$  i  $m$ . Funkcja ma zamienić miejscami drugi wiersz z przedostatnim. Jeśli tablica ma mniej niż cztery wiersze, funkcja ma nic nie robić. Stwórz przypadek testowy.

Przykład:

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

*Punktacja: 10 pkt.*

- Napisz strukturę `KontoBankowe` z polami `numerKonta` (tablica znaków długości 26) oraz `saldo` (typu `double`). Następnie napisz funkcję `wplacPieniadze`, której argumentem jest wskaźnik do struktury typu `KontoBankowe` oraz kwota, którą chcemy wpłacić. Funkcja ma dodać przekazaną kwotę do pola `saldo` w przekazanym argumencie. Stwórz przypadek testowy.

*Punktacja: 12 pkt.*

- Napisz funkcję, która przyjmuje jako argument listę bez głowy o elementach typu:

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

oraz liczbę całkowitą `a`. Funkcja wyświetlić te elementy listy, które są większa niż `a`. W przypadku braku takich elementów, funkcja ma nic nie robić. Stwórz jeden przypadek testowy.

*Punktacja: 22 pkt.*