

# Egzamin (drugi termin 2023)

## - Programowanie strukturalne - Zestaw M34

*Zadanie 1: 6 pkt. Zadanie 2: 12 pkt. Zadanie 3: 14 pkt. Zadanie 4: 18 pkt.*

*Punktacja: 46-50 pkt - bdb(5,0); 41-45 pkt - db+(4,5); 36-40 pkt - db(4,0); 31-35 pkt - dst+(3,5); 26-30 pkt - dst(3,0); 0-25 pkt - ndst (2,0).*

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`.
- 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).
- Kod zakomentowany nie będzie sprawdzany.

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

Polecenia są na odwrocie.

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 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).
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).

3. Zdefiniuj strukturę **Drzewo** z polami **gatunek** (tablica znaków długości 50) oraz **wysokosc** (typu **int**). Następnie napisz dwie funkcje:
  - a) **initDrzewo** - funkcja przyjmuje dwa argumenty: **gatunek** i **wysokość**, i zwraca wskaźnik na nowo utworzoną strukturę, ustawiając składowe z przekazanych argumentów. Dodatkowo funkcja powinna sprawdzić, aby **gatunek** był napisem długości co najmniej 2 i **wysokość** była większa niż 0. W przypadku nie spełnienia jednego z warunków, funkcja powinna zwracać **NULL**.
  - b) **zwiększWysokosc** - funkcja, której argumentem jest wskaźnik do struktury typu **Drzewo**. Funkcja ma dodać 5 do wysokości w przekazanym argumencie.

Upewnij się, że drugą funkcję możesz wywołać w **main**.

4. Napisz funkcję, która przyjmuje jako argument listę z głową o elementach typu:

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

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