

# Egzamin (drugi termin 2023)

## - Programowanie strukturalne - Zestaw M32

*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 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).
2. Napisz funkcję, która dostaje w argumencie napis. Funkcja ma usunąć z napisu wszystkie znaki będące cyframi (0-9). W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia. Stwórz przypadek testowy.
3. Zdefiniuj strukturę `KomputerStacjonarny` z polami `systemOperacyjny` (tablica znaków długości 50) oraz `ram` (typu `int`). Następnie napisz dwie funkcje i wywołaj każdą z nich co najmniej jeden raz:
  - a) `initKomputerStacjonarny` - funkcja przyjmuje dwa argumenty: system operacyjny i ram, i zwraca wskaźnik na nowo utworzoną strukturę, ustawiając składowe z przekazanych argumentów. Dodatkowo funkcja powinna sprawdzać, aby system operacyjny był napisem długości co najmniej 4 i ram był większy niż 0. W przypadku nie spełnienia jednego z warunków, funkcja powinna zwracać `NULL`.
  - b) `zwiększRam` - funkcja, której argumentem jest wskaźnik do struktury typu `KomputerStacjonarny`. Funkcja ma dodać 4 do ram w przekazanym argumencie.

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

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

```
struct ListNode {
    int s;
    struct ListNode * next;
};
```

oraz liczbę całkowitą `a`. Funkcja ma pomnożyć elementy podzielne przez 3 przez `a`. Stwórz jeden przypadek testowy.