

Przykładowe kolokwium - Zestaw 1

1. Napisz funkcję, która otrzymuje dwa napisy i dopisuje pierwszy na koniec drugiego. Zakładamy, że wynikowy napis zmieści się w pamięć zarezerwowanej dla tablicy przechowującej drugi napis. Stwórz jeden przypadek testowy dla funkcji.

Punktacja: 8 pkt.

2. Napisz funkcję, która porównuje dwie listy bez głowy o elementach typu:

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

i zwraca 1 jeżeli obie listy są równe (odpowiadające sobie elementy mają te same wartości pola x) oraz 0 w przeciwnym razie. Stwórz jeden przypadek testowy.

Punktacja: 10 pkt.

3. Dane są następujące wyrazy i znaki:

```
fun ** float ) tab int * n , unsigned ( float
```

Ułóż je we właściwej kolejności, aby otrzymać nagłówek funkcji `fun` przyjmującej w argumentach tablicę `tab` wskaźników na typ zmiennoprzecinkowy oraz jej długość `n`, zwracającej wskaźnik na typ zmiennoprzecinkowy.

Punktacja: 5 pkt.

4. Zdefiniuj strukturę składającą się z nazwy kraju, jego powierzchni w km² (liczba zmiennoprzecinkowa) oraz jego ludności (liczba całkowitoliczbowa). Napisz funkcję, która dla otrzymanej w pierwszym argumencie tablicy takich struktur, a w drugim — wielkości tablicy, zwróci strukturę o najmniejszym zagęszczeniu ludności wyrażonym wzorem: $\text{Zagęszczenie} = \text{ludność} / \text{powierzchnia}$. Stwórz jeden przypadek testowy dla funkcji.

Punktacja: 10 pkt.

5. Napisz funkcję, która przyjmie jako argumenty dwuwymiarową tablicę liczb zmiennoprzecinkowych, jej rozmiar `N` oraz liczbę zmiennoprzecinkową `X`. Funkcja ma za zadanie wypisać w kolejnych wierszach na standardowym wyjściu indeksy liczb większych od podanej liczby `X` oraz zwrócić liczbę tych liczb (ma to być zrealizowane w ramach jednej funkcji). Stwórz jeden przypadek testowy.

Punktacja: 12 pkt - jeśli rozwiązanie oparte jest na tablicach dynamicznych i wskaźnikach. 6 pkt - jeśli rozwiązanie oparte jest na tablicach o stałym rozmiarze. Punktowane jest tylko jedno z rozwiązań.

6. Przeanalizuj kod:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int r = 3, c = 4, i, j, count;
    int *arr[r];
    for (i=0; i<r; i++)
        arr[i] = (int *)malloc(c * sizeof(int));
    count = 0;
    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++)
            arr[i][j] = ++count;
    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++)
            if ((arr[i][j]+2)%3==0)
                printf("%d ", arr[i][j]);
    return 0;
}
```

Wpisz w poniższą tabelę to co zostanie wyświetlone na konsoli po wykonaniu programu. W jedną komórkę wpisz jeden znak.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Punktacja: 5 pkt.