Kolokwium II - Zestaw F

|  |  |
| --- | --- |
| Imię i nazwisko |  |
| Numer albumu |  |
| Punkty |  |  |  |  |  |  |  |

1. Napisz funkcję, która otrzymuje dwa napisy i dopisuje pierwszy na koniec drugiego w odwrotnej kolejności (zaczynając od ostatniego znaku drugiego napisu). Zakładamy, że wynikowy napis zmieści się w pamięć zarezerwowanej dla tablicy przechowującej odpowiednie napisy. Stwórz jeden przypadek testowy dla funkcji.

*Punktacja: 8 pkt.*

2. Napisz funkcję, która porównuje dwie listy z głową o elementach typu:

**struct** element {

 **float** x;

 **struct** element \* next;

};

i zwraca 1 jeżeli suma elementów na obu listach jest równa oraz 0 w przeciwnym razie. Stwórz jeden przypadek testowy.

*Punktacja: 8 pkt.*

3. Zdefiniuj typ wyliczeniowy roslina, mogący przyjmować wartości odpowiadające nazwom różnych roślin domowych. Stwórz jeden przypadek testowy dla typu wyliczeniowego.

*Punktacja: 5 pkt.*

4. Zdefiniuj strukturę Trapez, która przechowuje długości boków trapezu. Napisz funkcję, która przyjmuje jako parametr zmienna typu Trapez i zwraca obwód tak przekazanej figury. Stwórz przypadek testowy dla funkcji.

*Punktacja: 10 pkt.*

5. Napisz funkcję, która przyjmie jako argumenty dwuwymiarową tablicę liczb stałoprzecinkowych oraz jej wymiary m i n. Funkcja zwrócić iloczyn elementów stojących na miejscach o parzystych indeksach (zarówno dla wierszy i kolumn). Stwórz przypadek testowy dla funkcji.

Wskazówka: Dla poniższej tablicy funkcja powinna zwrócić $-18 = 2 ∙1∙9∙(-1)$.

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 2 | 1 | 2 |
| 7 | 7 | 4 | 4 |
| 9 | 9 | -1 | 1 |

*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() |  |  |  |  |  |  |  |  |
| { |  |  |  |  |  |  |  |  |
|  char wyraz[]="strukturalne"; |  |  |  |  |  |  |  |  |
|  printf("%p\n",wyraz); | 0 | 0 | 6 | 0 | F | E | C | 4 |
|  printf("%d\n", sizeof(char)); | 1 |   |   |   |   |   |   |   |
|  printf("%p\n",wyraz+5); |   |   |   |   |   |   |   |   |
|  printf("%c\n",\*wyraz+2); |   |   |   |   |   |   |   |   |
|  printf("%c\n",\*(wyraz+4)); |   |   |   |   |   |   |   |   |
|  printf("%c\n",(wyraz+6)[-2]); |   |   |   |   |   |   |   |   |
|  printf("%p\n",&(\*(wyraz+3))-1); |   |   |   |   |   |   |   |   |
|  return 0; |  |  |  |  |  |  |  |  |
| } |  |  |  |  |  |  |  |  |

*Punktacja: 7 pkt.*