Kolokwium II - Zestaw A

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

1. Napisz funkcję, która otrzymuje trzy napisy i dopisuje pierwszy na koniec drugiego, a drugi na koniec trzeciego. 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 {

 **char** 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: 8 pkt.*

3. Dane są następujące wyrazy i znaki:

\*\* char void int int int tab n m fun ( ) , , , sciezka \*

Ułóż je we właściwej kolejności, aby otrzymać nagłówek funkcji fun, która dostaje jako argumenty nazwę pliku, dwuwymiarową tablicę tablic o elementach typu int oraz wymiary tablicy.

*Punktacja: 5 pkt.*

4. Zdefiniuj strukturę planeta, posiadająca: numer planety (liczba całkowita), nazwa planety (napis 50-literowy), promień planety (liczba zmiennoprzecinkowa). Napisz funkcje przyjmującą tablicę planet oraz jej rozmiar. Funkcja ma zwrócić planetę (cała strukturę) o największej objętości ($V=\frac{4}{3}πr^{3}$). Stwórz jeden przypadek testowy dla funkcji. Wsk.: przyjmij za pi liczbę 3,14.

*Punktacja: 10 pkt.*

5. Napisz funkcję, która przyjmie jako argumenty dwuwymiarową tablicę liczb zmiennoprzecinkowych oraz jej rozmiary n i m. Funkcja ma zwrócić 1, jeśli na brzegach (tzn. w pierwszym i ostatnim wierszu oraz w pierwszej i ostatniej kolumnie) tablicy występują wartości niezerowe; w przeciwnym razie funkcja ma zwrócić 0.

*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 tab[3][4] = {{1,2,8,-2},{7,8,1,3},{-8,4,1,2}}; |  |  |  |  |  |  |  |  |
|  printf("%d\n",sizeof(int)); | 4 |   |   |   |   |   |   |   |
|  printf("%p\n",tab-1); | 0 | 0 | 6 | 0 | F | E | C | 0 |
|  printf("%d\n",\*(\*(tab+2)-1)); |   |   |   |   |   |   |   |   |
|  printf("%p\n",\*(tab+1)+3); |   |   |   |   |   |   |   |   |
|  printf("%d\n",\*(\*(tab+1)+4)); |   |   |   |   |   |   |   |   |
|  printf("%p\n",\*(tab-1)+7); |   |   |   |   |   |   |   |   |
|  printf("%p\n",\*tab+3); |   |   |   |   |   |   |   |   |
|  return 0; |  |  |  |  |  |  |  |  |
| } |  |  |  |  |  |  |  |  |

*Punktacja: 7 pkt.*