

Metody i języki programowania 4 sem, zaoczne

Zbiór zadań zaliczeniowych.

Na ocenę 3:

1. **Zera funkcji.** Napisać program znajdujący zera funkcji – wielomianu 3 stopnia metodą bisekcji. Wynik wpisz do pliku *wynik.txt*
2. **Generuj zakres.** Napisać program generujący zadaną ilość liczb z określonego przez użytkownika zakresu (np. od – 41 do 50). Liczby wpisz do pliku *wynik.txt*
3. **Kody rezystorów.** Napisać program przeliczający kody barwne rezystorów na liczby i odwrotnie.
4. **Zespolone.** Napisać program wykonujący działania na 2 liczbach zespolonych: dodawanie, odejmowanie, sprzężenie, mnożenie, dzielenie.
5. **Współrzędne.** Napisać program przeliczający współrzędne kartezjańskie na biegunowe i odwrotnie, wynik przeliczenia wpisz do pliku *wynik.txt*
6. **Tłumacz.** Napisać program tłumaczący liczbę całkowitą na tekst, np. (2003 = dwa tysiące trzy) dane liczbowe odczytywać z pliku, wynik tłumaczenia zapisać do pliku.
7. **Przelicznik.** Napisać program przeliczający miary długości z wybranych jednostek na różne jednostki. Wybrane 5 jednostek SI na inne.
8. **Pola figur.** Napisać program obliczający pola i objętości wybranych co najmniej 3 figur, brył.
9. **3D Minimalna odległość.** Napisać program, który w zbiorze 3D znajduje najmniejszą odległość. Współrzędna punktów pobierać z pliku. Współrzędne x , y , z każdego punktu znajdują się w pojedynczej linii, oddzielone są przecinkiem.
10. **2D Algebra.** Program obliczający w zbiorze 2D odległość 2 punktów, punktu od prostej, równanie prostej przechodzącej przez 2 punkty, rozwiązanie 2 równań liniowych.
11. **Stos / kolejka.** Program implementujący stos / kolejkę liczb całkowitych w postaci tablicy. Zdefiniuj funkcję *push* wstawiającą liczbę i *pop* zdejmującą liczbę oraz funkcję zapisującą stos do pliku.
12. **IX_to_9.** Program zamieniający liczby rzymskie na arabskie.
13. **Szyfr.** Program szyfrujący prostym szyfrem podstawieniowym pliki.

Na ocenę 4

1. **Systemy liczbowe.** Napisać program przeliczający całkowite liczby dziesiętne na system liczbowy 2, 8, 16, napisać funkcję przeliczającą liczby, nie używać dostępnych funkcji przeliczających języka C.
2. **Generator haseł.** Napisać program generujący hasła. Generować określoną ilość haseł o zadanej długości (ilości znaków), hasła składają się z liter lub cyfr – opcja.
3. **Zamiana napisów.** Napisać program zamieniający wybrany ciąg znaków występujący w pliku na inny ciąg znaków. Ciąg znaków nie zawiera znaków „białych”.
4. **Listing katalogu.** Napisać program, który wyświetla zawartość katalogu z możliwością użycia znaków blankietowych (jak polecenie dir). Jako 1 parametr programu podać znaki blankietowe oraz opcjonalnie 2 paramter - plik do którego wpisać listing. Wyświetlić czas utworzenia , datę, rozmiar, atrybuty.
5. **Zlicz napisy.** Napisać program zliczający występowanie w pliku danego zbioru ciągu znaków. Ciąg znaków nie zawiera znaków „białych”. Wykorzystaj funkcję strtok().
6. **Znajdź.** Napisać program zliczający występowanie w pliku słów kluczowych języka C. Wyświetl statystykę oraz zapisz ją do pliku.
7. **Sortuj liczby.** Napisz program sortujący liczby zapisane w pliku tekstowym. Tablicę przechowującą liczby utwórz dynamicznie. Opcja: sortuj rosnąco lub malejąco. Użyj funkcji qsort(). Posortowane liczby zapisz do pliku.

Na ocenę 5

1. **Sortuj.** Napisz program sortujący liczby rzeczywiste zapisane w pliku tekstowym. Posortowany zbiór wpisz do innego pliku. Format pliku: liczba w pierwszej linii określa ilość liczb do posortowania. Każda liczba zapisana jest w pojedynczej linii. Dynamicznie utwórz tablicę, w której przechowywane będą liczby odczytane z pliku. Napisz własną funkcję sortującą.
2. **Liczby z napisów.** Napisz program wstawiający liczby rzeczywiste z klawiatury do tablicy. Tablicę utworzyć dynamicznie na podstawie ilości liczb zadanej przez użytkownika. Z klawiatury odczytuj napisy (nie liczby). Do weryfikacji poprawności liczby użyj funkcji strtod(). Jeżeli napis jest niepoprawny, wyświetlić miejsce, od którego nastąpił błąd. Np. podajemy liczbę: 124k57 => błąd występuje od miejsca k57 – należy ten napis wyświetlić. Powtarzać aż do poprawnego wpisania przez użytkownika liczby. Wyświetl tablicę liczb.
3. **Struktura towar.** Zdefiniować strukturę zawierającą następujące pola: nazwa firmy, nazwa towaru, ilość sztuk, cena, jednostka monetarna.
 - a) napisz funkcję wypełniającą strukturę danymi z klawiatury,
 - b) napisz funkcję wyświetlającą dane,
 - c) napisz funkcję przeliczającą cenę na złote, dolary i marki,
 - d) napisz przykładowy program prezentujący działanie funkcji.

4. **Struktura macierz.** Zdefiniuj strukturę macierzy kwadratowej. Pamięć dla elementów macierzy alokuj dynamicznie.
 - a) napisz funkcję alokującą pamięć dla macierzy `matrix_malloc`
 - b) napisz funkcję zwalnającą pamięć `matrix_free`
 - c) napisz funkcję odczytującą macierz z pliku,
 - d) napisz funkcję zapisującą do pliku.
 - e) napisz przykładowy program prezentujący działanie wszystkich funkcji.

5. **Saper.** Gra „saper” (jak w systemie „Windows”)
 - a) utworzyć planszę 10 x 10,
 - b) wygenerować 10 min,
 - c) napisać funkcję zwracającą ilość min wokół wskazanego pola,

6. **Kółko / krzyżyk.** Gra w „kółko i krzyżyk”
 - a) wyświetlić planszę na ekranie.
 - b) opcje gry: użytkownik vs komputer, użytkownik vs użytkownik.

7. **Boolowskie.** Program zapisujący funkcję logiczną korzystając z tablicy prawdy max 4 zmiennych.

8. **Struktura student.** Zdefiniować strukturę `SStudent` zawierającą następujące pola: imię, nazwisko, numer indeksu, ocenę. Ustalić odpowiednie typy pól. W przypadku napisów użyj wskaźników. Napisz odpowiednie funkcje operujące na strukturze: modyfikacji danych, wyświetlania danych, zapisywania do pliku tekstowego, odczytywania z pliku tekstowego (ustal odpowiedni format pliku), inne niezbędne.

Sławomir Wróblewski
swroble@dmcs.p.lodz.pl