

Tematy zadań projektowych z Informatyki II

Uwaga: Programy z zad. 9-46 powinny działać poprawnie dla dowolnej ilości wczytywanych/losowanych liczb. Oznacza to, że nie możemy ustalić na stałe ilości liczb/znaków w kodzie. Może być ona podana z klawiatury, zapisana w pliku wejściowym lub program może sam się orientować ile jest liczb i dynamicznie zwiększać rozmiar tablicy. Należy rozważyć skorzystanie ze wskaźników i dynamicznej alokacji pamięci (funkcje *malloc* i *realloc* języka C lub operator *new[]* języka C++). Ewentualnie można skorzystać z biblioteki *vector*, jeśli program jest pisany w C++.

W nowszych standardach języka C (od C99) istnieje możliwość deklarowania tablic o zmiennym rozmiarze bez użycia dynamicznej alokacji pamięci – tzw. VLA - *variable length arrays*. VLA są stosunkowo proste w użyciu. Wystarczy zadeklarować tablicę o rozmiarze będącym zmienną. Nie są one jednak zalecane i mają swoje ograniczenia. Użycie tablic VLA zamiast dynamicznej alokacji jest dopuszczalne w programie, ale maksymalną możliwą oceną za taki projekt będzie 4.0.

Powyższa uwaga dotyczy jedynie zadań 9-46. Zadania 1-8 można wykonać dowolnym sposobem i nie wpłynie to na ocenę.

1. Napisz program, który wczytuje dwie macierze liczb całkowitych o rozmiarach 4x4 z osobnych plików do tablic dwuwymiarowych. Następnie program powinien obliczyć i wypisać na konsoli wyznaczniki obu macierzy, wykonać na nich operację mnożenia i zapisać wynikową macierz do osobnego pliku wyjściowego (w formacie takim samym, jak pliki wejściowe).

Poprawność programu można sprawdzić porównując wyniki do uzyskanych z innych tego typu aplikacji, np.: <https://matrixcalc.org/pl>

2. Napisz program, który wczytuje dwie macierze liczb zmiennoprzecinkowych (zmiennopozycyjnych) o rozmiarach 4x4 z osobnych plików do tablic dwuwymiarowych. Następnie program powinien obliczyć i wypisać na konsoli wyznaczniki obu macierzy, wykonać na nich operację dodawania i wynikową macierz transponować. Na koniec program powinien zapisać wynikową macierz do osobnego pliku wyjściowego (w formacie takim samym, jak pliki wejściowe).

Poprawność programu można sprawdzić porównując wyniki do uzyskanych z innych tego typu aplikacji, np.: <https://matrixcalc.org/pl>

3. Napisz program, który wczytuje z pliku następujące dane: x – początek przedziału (liczba zmiennoprzecinkowa), y – koniec przedziału (liczba zmiennoprzecinkowa), n – liczba punktów podziału. Program powinien obliczyć całkę oznaczoną z przedziału $[x - y]$ funkcji $f(x) = \sin(x) + \cos(x) + 2$. Należy napisać dwie wersje programu, wykorzystujące: (1) metodę prostokątów i (2) metodę Simpsona (https://eduinf.waw.pl/inf/alg/004_int/0001.php). Obie metody powinny być realizowane przez osobne funkcje. Należy zastanowić się która metoda zwraca dokładniejsze wyniki.

Poprawność programu można sprawdzić porównując wyniki do uzyskanych z innych tego typu aplikacji, np.: <https://online.etrapez.pl/kalkulatory/kalkulator-do-calek-oznaczonych-i-niewlasciwych>

4. Napisz program, który wczytuje z pliku następujące dane: x – początek przedziału (liczba zmiennoprzecinkowa), y – koniec przedziału (liczba zmiennoprzecinkowa), n – liczba punktów podziału. Program powinien obliczyć całkę oznaczoną z przedziału $[x - y]$ funkcji $f(x) = \sin(x) + \cos(x) + 1$. Należy napisać dwie wersje programu, wykorzystujące: (1) metodę trapezów i (2) metodę Simpsona (https://eduinf.waw.pl/inf/alg/004_int/0001.php). Obie metody powinny być realizowane przez osobne funkcje. Należy zastanowić się która metoda zwraca dokładniejsze wyniki.

Poprawność programu można sprawdzić porównując wyniki do uzyskanych z innych tego typu aplikacji, np.: <https://online.etrapez.pl/kalkulatory/kalkulator-do-calek-oznaczonych-i-niewlasciwych>

5. Napisz program, który wczytuje do tablicy *char* tekst o dowolnej długości zapisany w pliku. Następnie program powinien obliczać i wypisać na konsoli liczbę wszystkich znaków wraz z białymi (spacja, tabulacja, przejście do nowej linii), bez białych, liczbę cyfr, dużych liter, małych liter, znaków niebędących cyfrą ani literą, samogłosek i spółgłosek. Program powinien również przygotować wczytany tekst z zamienionymi wszystkimi dużymi literami na małe i zapisać go w osobnym pliku. Zamiana liter dużych na małe powinna być zaimplementowana jako funkcja, którą program wywołuje.

Program może uwzględniać litery z podstawowego angielskiego alfabetu. Nie musi uwzględniać polskich liter typu „ą”, „ę”.

6. Napisz program, który wczytuje z pliku liczbę k oraz wielkość zbioru n (liczby całkowite). Następnie program powinien obliczyć i zapisać w osobnym pliku następujące wartości kombinatoryczne:

- liczbę k -elementowych wariacji z powtórzeniami ze zbioru n -elementowego,
- liczbę k -elementowych wariacji bez powtórzeń ze zbioru n -elementowego,
- liczbę k -elementowych kombinacji z powtórzeniami ze zbioru n -elementowego,
- liczbę k -elementowych kombinacji bez powtórzeń ze zbioru n -elementowego.

Każda z powyższych wartości powinna być obliczana przez osobną funkcję.

7. Napisz program, który losuje liczbę k oraz wielkość zbioru n (liczby całkowite). Przedziały losowanych liczb n i k powinny być zadane przez użytkownika poprzez konsolę. Następnie program powinien obliczyć i zapisać w pliku wyjściowym następujące wartości kombinatoryczne:

- liczbę k -elementowych wariacji z powtórzeniami ze zbioru n -elementowego,
- liczbę k -elementowych wariacji bez powtórzeń ze zbioru n -elementowego,
- liczbę k -elementowych kombinacji z powtórzeniami ze zbioru n -elementowego,
- liczbę k -elementowych kombinacji bez powtórzeń ze zbioru n -elementowego.

Każda z powyższych wartości powinna być obliczana przez osobną funkcję.

8. Napisz program, który losuje współczynniki a , b i c (liczby zmiennoprzecinkowe) funkcji kwadratowej ($f(x) = ax^2 + bx + c$). Przedział losowanych wartości współczynników powinien być zadany przez użytkownika poprzez konsolę. Następnie program powinien obliczyć miejsca zerowe funkcji w dziedzinie liczb rzeczywistych. W pliku wyjściowym należy zapisać dla każdej funkcji:

- obliczone miejsca zerowe, jeśli istnieją;
- informację o braku miejsc zerowych, jeśli delta jest ujemna;
- informację, że funkcja nie jest kwadratowa, jeśli współczynnik a jest zerowy.

Pierwiastek kwadratowy można obliczyć za pomocą funkcji `sqrt` z biblioteki `math.h` (język C) lub `cmath` (język C++).

9. Napisz program, który wczytuje współczynniki a , b i c funkcji kwadratowych ($f(x) = ax^2 + bx + c$) z pliku do tablic. Każda linia pliku powinna odpowiadać jednej funkcji. Następnie program powinien obliczyć miejsca zerowe funkcji w dziedzinie liczb rzeczywistych. W osobnym pliku należy zapisać dla każdej funkcji:

- obliczone miejsca zerowe, jeśli istnieją;
- informację o braku miejsc zerowych, jeśli delta jest ujemna;
- informację, że funkcja nie jest kwadratowa, jeśli współczynnik a jest zerowy.

Pierwiastek kwadratowy można obliczyć za pomocą funkcji `sqrt` z biblioteki `math.h` (język C) lub `cmath` (język C++).

Program powinien działać poprawnie dla dowolnej ilości wczytywanych funkcji.

10. Napisz program, który wczytuje kolejne dodatnie liczby zmiennoprzecinkowe (zmiennopozycyjne) z pliku do tablicy. Następnie program powinien obliczyć następujące wartości statystyczne wczytanych liczb: średnia arytmetyczna, średnia geometryczna, średnia harmoniczna, wariancja i odchylenie standardowe. Obliczone wartości należy zapisać w osobnym pliku.

Pierwiastek dowolnego stopnia można obliczyć za pomocą funkcji potęgowania *pow* z biblioteki *math.h* (język C) lub *cmath* (język C++). Należy przypomnieć sobie jak według zasad matematyki można obliczyć pierwiastek za pomocą potęgowania.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

11. Napisz program, który losuje n dodatnich liczb zmiennoprzecinkowych (zmiennopozycyjnych) z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie program powinien obliczyć następujące wartości statystyczne wczytanych liczb: średnia arytmetyczna, średnia geometryczna, średnia harmoniczna, wariancja i odchylenie standardowe. Obliczone wartości należy zapisać w osobnym pliku.

Pierwiastek dowolnego stopnia można obliczyć za pomocą funkcji potęgowania *pow* z biblioteki *math.h* (język C) lub *cmath* (język C++). Należy przypomnieć sobie jak według zasad matematyki można obliczyć pierwiastek za pomocą potęgowania.

12. Napisz program, który wczytuje kolejne dodatnie liczby zmiennoprzecinkowe (zmiennopozycyjne) oraz ich wagi z pliku do tablicy. Następnie program powinien obliczyć w osobnych funkcjach następujące wartości statystyczne wczytanych liczb: średnia arytmetyczna, ważona średnia arytmetyczna i odchylenie przeciętne. Obliczone wartości należy zapisać w osobnym pliku.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

13. Napisz program, który losuje n dodatnich liczb zmiennoprzecinkowych (zmiennopozycyjnych) oraz ich wag z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie program powinien obliczyć w osobnych funkcjach następujące wartości statystyczne wczytanych liczb: średnia arytmetyczna, ważona średnia arytmetyczna i odchylenie przeciętne. Obliczone wartości należy zapisać w osobnym pliku.

14. Zdefiniuj strukturę *Samochod* posiadającą pola: *marka* typu *char**, *rok_produkcji* typu *int* i *cena* typu *float*. Napisz program, który wczytuje dane kolejnych samochodów z pliku do tablicy struktur. Następnie program powinien obniżyć cenę wszystkich samochodów wyprodukowanych między 2016 a 2021 r. o 10% i uaktualnić dane w pliku wejściowym (nadpisać poprzednie dane nowymi).

Program powinien działać poprawnie dla dowolnej liczby wczytywanych samochodów.

15. Zdefiniuj strukturę *Pracownik* posiadającą pola: *imie* i *nazwisko* typu *char**, *rok_zatrudnienia* typu *int* i *pensja* typu *float*. Napisz program, który wczytuje dane kolejnych pracowników z pliku do tablicy struktur. Następnie program powinien przyznać podwyżkę wszystkim pracownikom zatrudnionym między 2015 a 2020 r. o 15% i uaktualnić dane w pliku wejściowym (nadpisać poprzednie dane nowymi).

Program powinien działać poprawnie dla dowolnej liczby wczytywanych pracowników.

16. Zdefiniuj strukturę *Punkt* posiadającą pola: *symbol* typu *char*, *x* i *y* typu *float*. Pierwsze pole oznacza symbol punktu, np. 'A', 'B', 'C', natomiast dwa ostatnie pola oznaczają jego współrzędne. Napisz program, który wczytuje dane kolejnych punktów z pliku do tablicy struktur. Następnie program powinien obliczyć i zapisać w osobnym pliku odległości euklidesowe pomiędzy wszystkimi punktami. Wzór na odległość euklidesową znajduje się tutaj:

https://www.statsoft.pl/textbook/stathome_stat.html?https%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstcluan.html

Pierwiastek kwadratowy można obliczyć za pomocą funkcji *sqrt* z biblioteki *math.h* (język C) lub *cmath* (język C++).

Program powinien działać poprawnie dla dowolnej liczby wczytywanych punktów.

17. Zdefiniuj strukturę *Punkt* posiadającą pola: *symbol* typu *char*, *x* i *y* typu *float*. Pierwsze pole oznacza symbol punktu, np. 'A', 'B', 'C', natomiast dwa ostatnie pola oznaczają jego współrzędne. Napisz program, który wczytuje dane kolejnych punktów z pliku do tablicy struktur. Następnie program powinien obliczyć i zapisać w osobnym pliku odległości miejskie pomiędzy wszystkimi punktami. Wzór na odległość miejską znajduje się tutaj:

https://www.statsoft.pl/textbook/stathome_stat.html?https%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstcluan.html

Program powinien działać poprawnie dla dowolnej liczby wczytywanych punktów.

18. Zdefiniuj strukturę *Grupa* oraz *Student*. Struktura *Grupa* powinna mieć pola *wielkosc* (liczba całkowita) i *studenci* (tablica obiektów *Student*). Struktura *Student* powinna mieć pola *imie* i *nazwisko* (napisy/tablice znaków). Następnie napisz program, który

wczytuje z pliku przykładowe dwie grupy studentów. Następnie należy wypisać na konsoli wczytane dane.

Do wczytywania imion i nazwisk można użyć funkcji *fgets* z biblioteki *stdio.h* (język C) lub *cstdio* (język C++).

Program powinien działać poprawnie dla dowolnej liczby wczytywanych studentów w grupach i dla dowolnej długości imion i nazwisk.

19. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Napisz również funkcję realizującą algorytm wyszukiwania binarnego. Następnie, program powinien poprosić użytkownika o podanie liczby, którą chce wyszukać, wywołać funkcję wyszukiwania binarnego i zwrócić jej wynik. Jeśli znaleziono liczbę, to program powinien wypisać jej indeks, a w przeciwnym wypadku powinien wypisać informację „nie znaleziono”.

Należy pamiętać, że wyszukiwanie binarne działa jedynie na posortowanej tablicy liczb.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

20. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Napisz również funkcję realizującą algorytm wyszukiwania interpolacyjnego. Następnie, program powinien poprosić użytkownika o podanie liczby, którą chce wyszukać, wywołać funkcję wyszukiwania binarnego i zwrócić jej wynik. Jeśli znaleziono liczbę, to program powinien wypisać jej indeks, a w przeciwnym wypadku powinien wypisać informację „nie znaleziono”.

Należy pamiętać, że wyszukiwanie interpolacyjne działa jedynie na posortowanej tablicy liczb.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

21. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Następnie, napisz funkcję, która znajduje liczbę największą, najmniejszą, najczęściej powtarzającą się (https://eduinformatyka.waw.pl/inf/alg/001_search/0036.php) oraz liczbę jej powtórzeń i zapisuje te dane do osobnego pliku wyjściowego.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

22. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania bąbelkowego w wersji zoptymalizowanej (https://eduinformatyka.waw.pl/inf/alg/003_sort/0007.php) i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

23. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez wstawianie i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

24. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez wybieranie (przez wybór) i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

25. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez zliczanie i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

26. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez scalanie i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

27. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania kubekowego i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

28. Napisz program, który wczytuje kolejne liczby całkowite z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania szybkiego i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

29. Napisz program, który losuje n liczb całkowitych z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania bąbelkowego w wersji

zoptymalizowanej (https://eduinf.waw.pl/inf/alg/003_sort/0007.php) i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

30. Napisz program, który losuje n liczb całkowitych z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, napisz i wywołaj funkcję, która znajduje liczbę największą, najmniejszą, najczęściej powtarzającą się (https://eduinf.waw.pl/inf/alg/001_search/0036.php) oraz liczbę jej powtórzeń i zapisuje te dane do pliku wyjściowego.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

31. Napisz program, który losuje n liczb całkowitych z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez wstawianie i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

32. Napisz program, który losuje n liczb całkowitych z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez wybieranie (przez wybór) i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

33. Napisz program, który losuje n liczb całkowitych z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez zliczanie i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

34. Napisz program, który losuje n liczb całkowitych z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez scalanie i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

35. Napisz program, który losuje n liczb całkowitych z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania kubełkowego i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

36. Napisz program, który losuje n liczb całkowitych z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy na przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania szybkiego i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

37. Napisz program, który wczytuje kolejne liczby zmiennoprzecinkowe (zmiennopozycyjne) z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania bąbelkowego w wersji zoptymalizowanej (https://eduinf.waw.pl/inf/alg/003_sort/0007.php) i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

38. Napisz program, który wczytuje kolejne liczby zmiennoprzecinkowe (zmiennopozycyjne) z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez wstawianie i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

39. Napisz program, który wczytuje kolejne liczby zmiennoprzecinkowe (zmiennopozycyjne) z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez wybieranie (przez wybór) i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

40. Napisz program, który wczytuje kolejne liczby zmiennoprzecinkowe (zmiennopozycyjne) z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez scalanie i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

41. Napisz program, który wczytuje kolejne liczby zmiennoprzecinkowe (zmiennopozycyjne) z pliku do tablicy. Następnie, program powinien posortować liczby algorytmem sortowania szybkiego i zapisać je w osobnym pliku wyjściowym, w formacie takim samym, jak plik wejściowy. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.

Program powinien działać poprawnie dla dowolnej ilości wczytywanych liczb.

42. Napisz program, który losuje n liczb zmiennoprzecinkowych (zmiennopozycyjnych) z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania bąbelkowego w wersji zoptymalizowanej (https://eduinf.waw.pl/inf/alg/003_sort/0007.php) i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.
43. Napisz program, który losuje n liczb zmiennoprzecinkowych (zmiennopozycyjnych) z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez wstawianie i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.
44. Napisz program, który losuje n liczb zmiennoprzecinkowych (zmiennopozycyjnych) z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez wybieranie (przez wybór) i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.
45. Napisz program, który losuje n liczb zmiennoprzecinkowych (zmiennopozycyjnych) z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania przez scalanie i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.
46. Napisz program, który losuje n liczb zmiennoprzecinkowych (zmiennopozycyjnych) z przedziału zamkniętego $[x - y]$ (n , x oraz y podajemy przez konsolę z klawiatury) i zapisuje je w tablicy. Następnie, program powinien posortować liczby algorytmem sortowania szybkiego i zapisać je w pliku wyjściowym. Sortowanie powinno być zaimplementowane jako funkcja, którą program wywołuje.