Programowanie proceduralne lab 5

Zadanie 1. (4 pkt)

Metoda „sito Eratostenesa" służy do wykrywania liczb pierwszych. Pozwala znaleźć wszystkie liczby pierwsze mniejsze od zadanej wartości n.
Polega to na utworzeniu listy liczb całkowitych od 1 do n i wykreślaniu liczb podzielnych przez inne.

Algorytm:
1.Wykreśla się liczbę 1,
2. Poszukuje się, poczynając od ostatniej znalezionej liczby pierwszej (za pierwszym razem od 1) najbliższej niewykreślonej liczby. Liczba to jest pierwsza. Następnie wykreśla się z listy wszystkie liczby podzielne przez tę liczbę pierwszą (tzn. np. zerujemy jej kolejne wielokrotności w tablicy)
3. powtarza się krok dwa aż do chwili, gdy znaleziona liczba pierwsza będzie większa od $N^{1/2}$
można dowieść, że w tym momencie wszystkie liczby <N nie będące pierwszymi zostały wykreślone z listy

Napisz funkcję która będzie deklarować dynamicznie tablicę znaków o zadanym rozmiarze, wypełniać ją np. znakiem jeden '1' (użyj funkcji memset())i zwracać do niej wskaźnik. Indeksy elementów tablicy to kolejne liczby całkowite więc nie trzeba do tablicy wpisywać ich wartości). Skreślenie liczby, która nie jest pierwsza może polegać na wpisanie do elementu tablicy o tym indeksie znaku '0'

Wykorzystując powyższy algorytm napisz program który szuka liczb pierwszych mniejszych od podanej przez użytkownika wartości N. Utwórz dynamicznie tablicę, która pozwoli wyznaczyć liczby pierwsze. Sprawdź ile liczb udało Ci się odszukać i wypisz je oraz wypisz ich ilość. Utwórz tablicę, która będzie zawierała same liczby pierwsze z zadanego przedziału.

Zadanie 2. (3 pkt)

Dany jest przykładowy schemat programu, uwaga! funkcja main() zawiera argumenty wywołania:

```
#include...
int main(int argc, char *argv[]) // (argument count, argument values)
int main(int argc, char **argv) // postać alternatywna
{
//1.  Wypisz ile jest argumentów (argc)

//2.W pętli wypisz argumenty
//printf("%d: %s\n",arg_no, argv[arg_no])

//3. dokonaj konwersji łańcuchów znaków na liczby przy pomocy funkcji:
atoi() (alphanumeric to integer), atof(), atol()

np.liczba= atoi(argv[1]))
```

Jeśli argument nie jest rozpoznawalny jako liczba funkcja zwraca wartość 0 (umieść stosowny komunikat)

A. Wykonaj test dla przykładowych argumentów :
np. ./a.out jeden 2 4.5

```
 int main(int argc, char **argv){

//Wypisanie argumentów:
printf("argc = %d\n", argc); // ilość argumentów
for(int i=0; i<argc; i++){
printf("argv[%d] = %s\n", i, argv[i]);
 }

 //2. Konwersja łańcucha znaków na zmienne typu int i float

int integerNumber = atoi(argv[1]);
float floatNumber = atof(argv[2]);
printf("Integer: %d, Float: %0.2f\n", integerNumber, floatNumber);
```

B. Utwórz tablicę 80 znaków wczytaj do niej dowolny podany z klawiatury łańcuch znaków wypisz na ekran. Użyj funkcji : scanf () , drugi raz gets() , trzeci raz fgets()
Wpisz ten łańcuch znaków do pliku znaki.txt wykonaj teraz program przekierowując strumień wejściowy stdin w następujacy sposób a.out <znaki.txt
i wczytaj łańcuch używając funkcji scanf() a potem fgets()

Zadanie 3. (3 pkt)

Wykorzystując argumenty wywołania funkcji main() zadaj rozmiary 2D tablicy w wierszu poleceń, oraz zakres liczb losowych, którymi wypełnisz tablicę .
Dokonaj dynamicznej alokacji pamięci i sprawdź czy alokacja została dokonana prawidłowo (jeżeli nie, zakończ działanie programu).
Utwórz tablicę i wypełnij ją liczbami losowymi z zadanego przedziału.
Napisz funkcję która będzie wykonywała sortowanie tablicy, oraz funkcję wykonująca normalizację tablicy tak by suma wszystkich elementów była równa 1.