**Zadanie 1 (10 pkt. na pracowni, później 5 pkt.)**

Napisz program, który wypisuje na standardowe wyjście wszystkie możliwe wartościowania zbioru VARCNT zmiennych zdaniowych w zbiorze wartości logicznych {T, F}. Na przykład dla VARCNT = 3 będzie to

T T T

T T F

T F T

T F F

F T T

F T F

F F T

F F F

Wartość VARCNT ustaw przy pomocy dyrektywy #define na jakąś rozsądną wartość (np. 6, a z całą pewnością nie więcej niż 32). To zadanie można rozwiązać na (co najmniej) dwa sposoby – wykorzystując funkcje rekurencyjne bądź operacje bitowe i oczywiście każde będzie równie dobre (a dla własnej satysfakcji najlepiej zaimplementuj oba).

**Zadanie 2 (10 pkt.)**

Napisz funkcję int nastepne(char wzorzec[], char tekst[], int poczatkowa) zwracającą pozycję w łańcuchu tekst nie wcześniejszą niż poczatkowa, na której występuje wzorzec, bądź -1, jeśli takiego wystąpienia nie ma. Wywołanie nastepne("ala", "kalamarz", n) powinno zwracać 1 dla n = 0 bądź 1, oraz -1 dla większych wartości n.

Napisz funkcję void raportuj(char wzorzec[], char tekst[], int nr_linii) wykorzystującą poprzednią funkcję do wypisania na standardowe wyjście wiersza zawierającego kolejno nr_linii, dwukropek, oraz oddzielone spacjami pozycje, na których w łańcuchu tekst występuje wzorzec. Jeśli takich wystąpień nie ma, to nie powinno być wypisywane nic (nawet pusty wiersz).

Napisz program wykorzystujący powyższe funkcje, który w wierszu poleceń przyjmuje dokładnie jeden argument wywołania niebędący pustym napisem (w przeciwnym przypadku natychmiast kończy działanie, wypisując na wyjście zrozumiały komunikat o zaistniałej sytuacji) zawierający wzorzec, czyta standardowe wejście i wypisuje pozycje, na których występuje w nim wzorzec. Wystąpienia wzorca mają być wykrywane bez uwzględnienia wielkości liter (ASCII) – do Twojej decyzji należy, czy odpowiednio dostosujesz funkcję nastepne, czy rozwiążesz to inaczej. Zasadnicza część funkcji main powinna wyglądać mniej więcej tak:

*for (int i = 0; fgets(s, MAXLEN, stdin) != NULL; i++) raportuj(wz, s, i);*

Wartość MAXLEN ustaw przy pomocy dyrektywy #define na jakąś rozsądną wartość (np. 10000). Pamiętaj, że łańcuchy w C (w tym też te w argv czy zwracane przez fgets) kończą się znakiem 0 ('\0') i uwzględnij to w swoich implementacjach.

**Zadanie 3 (ze sprawdzaczką, 10 pkt.):**

Permutacja n-elementowego zbioru {1, 2, …, n} to wzajemnie jednoznaczne przekształcenie tego zbioru na siebie (bijekcja). Liczba wszystkich permutacji n-elementowego zbioru wynosi n!. Rozważmy dla przykładu pewną permutację σ zbioru {1, 2, 3, 4, 5, 6, 7}, którą zapiszemy w postaci tabelarycznej:

| **i** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| **σ(i)** | 4 | 5 | 3 | 7 | 2 | 1 | 6 |

Często permutacje zapisuje się w formie skróconej w postaci ciągu wartości przypisanych kolejnym liczbom (drugi wiersz tabeli), co dla powyższego przykładu wygląda tak:

*4, 5, 3, 7, 2, 1, 6*

Oznacza to, że permutacja dokonuje następujących przekształceń:

*1→4, 2→5, 3→3, 4→7, 5→2, 6→1, 7→6*

Porządkując te przekształcenia otrzymamy następujące cykle:

*1→4, 4→7, 7→6, 6→1; 2→5, 5→2; 3→3*

Widać więc, że permutację można przedstawić w postaci rozłącznych cykli:

(1, 4, 7, 6); (2, 5); (3)

Napisz program znajdujący długość najdłuższego cyklu zadanej permutacji.