

Laboratorium nr 5

Tablice, łańcuchy i wskaźniki

Informatyka

Po zakończeniu laboratorium wszystkie zrobione zadania należy przesłać pocztą elektroniczną do prowadzącego zajęcia!

Zadania

1. Zliczyć ilość wystąpień wzorca w ciągu bazowym (oba są ciągami znakowymi).
2. Wypisać znak, który występuje największą ilość razy w znakowym ciągu wejściowym.
3. Wyznaczyć najdłuższy podciąg, wejściowego ciągu znakowego składający się z tych samych znaków zdefiniowanych na wejściu.
4. Napisz program, który w zadanym (nieposortowanym) ciągu n liczb zidentyfikuje wszystkie inwersje. Inwersje należy policzyć i wylistować. Niech $A[1..n]$ będzie tablicą zawierającą n różnych liczb. Jeśli $i < j$ oraz $A[i] > A[j]$, to para (i,j) jest nazywana inwersją A . Na przykład w ciągu: 1,2,5,4,7,12,23,125,41,78,130 inwersjami są pary: (5,4), (125,41) oraz (125,78).
5. Napisz program, który realizuje następujące funkcje dla tablic jednowymiarowych:
 - Dodaje odpowiadające sobie elementy tablic – suma dwu wektorów,
 - Mnoży odpowiadające sobie elementy tablic,
 - Wyznacza różnicę pomiędzy maksymalnym a minimalnym elementem tablicy,
6. Zaimplementować funkcję pozwalającą na zamianę dwu wybranych wierszy macierzy, wykorzystaj wskaźniki.
7. Ćwiczenie w wykorzystaniu zapisu wskaźnikowego jako metody dostępu do elementów tablicy. Napisz funkcje wykonujące podstawowe operacje na tablicach (tzn. wczytywanie, wyświetlanie, liczenie średniej, max, min) ale w zapisie wskaźnikowym.
8. Napisz funkcje:

```
int moje_strlen( char* );
char* moje_strupr(char* );
char* moje_strcpy( char*, char* )
```

realizujące te same operacje co standardowe funkcje `strlen`, `strupr`, `strcpy` z biblioteki `<string.h>`.
9. Napisz funkcję:

```
void str_zamien( char* tekst, char* stary_wzorzec, char* nowy_wzorzec);
```

realizującą operację zamiany w podanym tekście starego wzorca na nowy np. `<tekst> == "ala ma kota a ola ma asa"` `<stary_wzorzec> == "ma"` `<nowy_wzorzec> == "miała"` rezultat zamiany: "ala miała kota a ola miała asa"

(Osoby mniej wprawne w programowaniu mogą napisać uproszczoną wersję tej funkcji, zamieniając tylko wzorce o tej samej długości. Wówczas nie ma potrzeby "rozsuwania" lub "zsuwania" starego tekstu)

10. Napisz program szyfrujący systemem Juliusza Cezara używanym w starożytnym Rzymie (kodowanie i rozkodowanie). System jest następujący: założmy, że mamy 26-literowy alfabet A-Z oraz jego liczbowe odpowiedniki 0-25. Niech P należące do zbioru $\{0,1,2,\dots,25\}$ oznacza jednostkę tekstu jawnego, natomiast C - jednostkę kryptogramu. Zdefiniujmy funkcję $f(\{0,1,2,\dots,25\}) \rightarrow \{0,1,2,\dots,25\}$ za pomocą wzoru: $C = f(P) = P+3$ dla $x < 23$ oraz $C = f(P) = P-23$ dla $x \geq 23$. Inaczej mówiąc f dodaje 3 modulo 26. Uzyskane liczby zamieniamy z powrotem na litery. Rozszyfrowanie jest funkcją odwrotną i polega na odejmowaniu 3 modulo 26 ($P = f^{-1}(C) = C-3 \pmod{26}$). Ten system kryptograficzny można uogólnić: dla danego alfabetu N -literowego oraz zbioru liczb $\{0,1,2,\dots,N-1\}$, ustalamy klucz szyfrujący b oraz funkcję f zwaną przesunięciem: $f(P) = P-b \pmod{N}$. Funkcja odwrotna: $f^{-1}(C) = C-b \pmod{N}$.

Przykład. Zszyfrujmy teraz w naszym systemie kryptograficznym słowo „YES”. Najpierw przekształcamy litery słowa na liczby: 24 4 18, potem dodajemy 3 modulo 26: 1 7 21, a na końcu z powrotem przekształcamy na litery: „BHV”.