

24. Realizacja algorytmu, czyli program szuka wzorca

NA TEJ LEKCJI:

- ułożysz program wyszukujący wzorzec w teście.

Znasz już metodę naiwną wyszukiwania wzorca w tekście. Wiesz, jak działa przykładowy algorytm. Na jego podstawie można ułożyć skuteczny program. Oczywiście problem zakodowania algorytmu można rozwiązać na wiele sposobów. Programy mogą się różnić nie tylko sposobem wyświetlania wyniku, lecz także zastosowaniem różnych instrukcji i rozkazów. Zapoznaj się z dwoma rozwiązaniami.

24.1. Według przepisu, czyli układamy program realizujący przykładowy algorytm

założenia

Ponieważ nasz program ma głównie realizować algorytm, uprościmy go, z góry zakładając maksymalną długość tekstu (`dlTekstu=100`) i wzorca (`dlWzorca=100`). Dodajmy za to ciekawy sposób wyświetlania wyniku (rys. 24.1.). Użyjemy w tym celu strumienia `cout`.

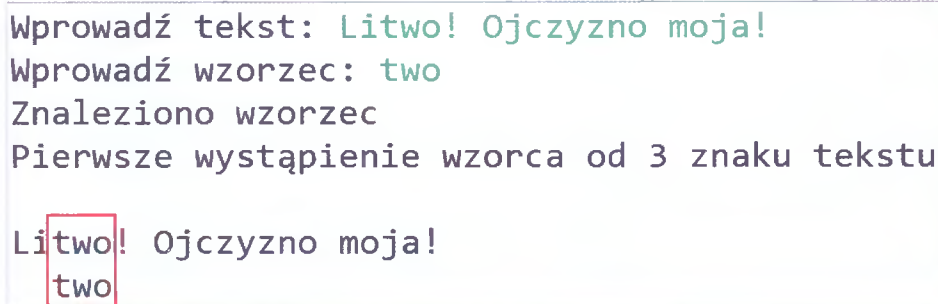
`cout.width`

Zastosowanie `cout.width(n+dlWzorca)` pozwoli określić, ile znaków łącznie będzie zawierał strumień. Dzięki temu wzorzec zostanie wyświetlony w odpowiednim miejscu (rys. 24.1.). Uwzględnia białe znaki, które są niewidoczne na ekranie. Przykładem może być spacja.

białe znaki

`cin.getline`

Dzięki użyciu `cin.getline` będzie możliwe wczytanie tekstu lub wzorca do momentu użycia entera. Tablice `t[]` i `w[]` zostaną wypełnione wprowadzonym tekstem.



```

Wprowadź tekst: Litwo! Ojczyzno moja!
Wprowadź wzorzec: two
Znaleziono wzorzec
Pierwsze wystąpienie wzorca od 3 znaku tekstu

Litwo! Ojczyzno moja!
two
  
```

Wzorzec jest wyświetlany w miejscu pierwszego wystąpienia w teście.

Rys. 24.1. Przykładowy sposób wyświetlania wyniku działania programu wyszukującego wzorzec w teście



funkcja
strlen()

W programie zastosujemy także funkcję `strlen()` (zawartą w bibliotece `string.h`), która oblicza (zwraca) liczbę znaków tablicy przekazanej w nawiasie jako jej argument. Pozwoli to na ustalenie długości tekstu (`t[]`) i wzorca (`w[]`).

ograniczenia
w programie

```

1 #include <iostream>
2 #include <string.h>
3 using namespace std;
4
5 int main()
6 {
7     char t[100], w[100];
8     int n=0;
9     cout << "Wprowadź tekst: ";
10    cin.getline(t, 100);
11    cout << "Wprowadź wzorzec: ";
12    cin.getline(w, 100);
13
14    int dlTekstu = strlen(t);
15    int dlWzorca = strlen(w);
16
17    while (n<=dlTekstu-dlWzorca)
18    {
19        int j = 0;
20        while ((j<dlWzorca) && w[j]==t[n + j])
21            j=j+1;
22        if (j==dlWzorca)
23        {
24            cout << "Znaleziono wzorzec" << endl;
25            cout << "Pierwsze wystąpienie wzorca od " << n + 1 << " znaku tekstu" << endl << endl;
26            cout << t << endl;
27            cout.width(n+dlWzorca);
28            cout << w << endl;
29            return 0;
30        }
31        n=n+1;
32    }
33    cout << "nie znaleziono wzorca";
34    return 0;
35 }
36

```

Znaleziono wzorzec
Pierwsze wystąpienie wzorca od 3 znaku tekstu

Litwo! Ojczyzno moja!
two

Rys. 24.2. Przykładowy program poszukujący wzorca w tekście

ZADANIA DO ROZWIĄZANIA

1. Czy program z rysunku 24.2. będzie działał poprawnie, gdy wprowadzisz wzorzec dłuższy od tekstu? Uzasadnij swoją odpowiedź. Sprawdź ją doświadczalnie dla różnych danych.
2. Uzasadnij nadanie początkowych wartości 0 zmiennym n oraz j .
3. Opisz rolę zmiennej j w programie z rysunku 24.2.
- 4*. Ułóż program różniący się sposobem realizacji metody naiwnej wyszukiwania wzorca w tekście od przedstawionego na rysunku 24.2. Opisz jego działanie.
- 5*. Zmodyfikuj program z rysunku 24.2. w taki sposób, by odnaleziony wzorzec był wskazany w teście kolorem czerwonym. Pomiń wyświetlanie wzorca pod tekstem.

PODSUMOWANIE LEKCJI

- s. 182 • Dla uproszczenia możesz przyjąć stałą wielkość tablic `t []` oraz `w []`.
- s. 182 • `cout.width` określi, ile znaków łącznie będzie zawierał strumień.
- s. 182 • Dzięki użyciu `cin.getline` będzie możliwe wprowadzenie tekstu lub wzorca do momentu użycia entera.
- s. 182 • Białe znaki są znakami niewidocznymi na ekranie.
- s. 182 • Sposób wyświetlania efektu działania programu może wskazywać jednocześnie miejsce w tekście, w którym znaleziono wzorzec.
- s. 183 • Funkcja `strlen()` (zawarta w bibliotece `string.h`) oblicza (zwraca) liczbę znaków tablicy przekazanej w nawiasie jako jej argument.
- s. 183 • Przykładowy program ma ograniczenia, np. liczbę znaków w tekście lub wzorcu.