T: Wyszukiwanie wzorca w tekście.

1. **Wyszukiwanie wzorca** w tekście polega na znalezieniu wszystkich wystąpień określonego łańcucha znaków w tekście. Wynikiem jest ciąg liczb wskazujących pozycje w przeszukiwanym tekście, od których rozpoczynają się wystąpienia szukanego wzorca.

Przykład

Przeanalizujmy słowo „KARYKATURA". Załóżmy, że szukanym wzorcem jest tekst „KA". Uzyskany wynik to liczby 0 i 4, ponieważ od tych pozycji rozpoczyna się wystąpienie wzorca „KA".

0	1	2	3	**4**	5	6	7	8	9
K	**A**	R	Y	**K**	**A**	T	U	R	A

Szarym tłem zaznaczono pozycje określające położenie wzorca w tekście, natomiast niebieskim kolorem wyróżniono wzorzec występujący w podanym słowie.

2. **Algorytm naiwny** realizujący wyszukiwanie wzorca w tekście opiera się na metodzie przeszukiwania liniowego. Tekst przeglądany jest znak po znaku, począwszy od pierwszej litery.

Przykład

W przypadku tekstu „KARYKATURA" i wzorca „KA" przebieg algorytmu jest następujący:

0	1	2	3	**4**	5	6	7	8	9
K	**A**	R	Y	**K**	**A**	T	U	R	A
K	**A**								
	K	A							
		K	A						
			K	A					
				K	**A**				
					K	A			
						K	A		
							K	A	
								K	A

Szarym tłem wyróżniono pozycje określające położenie wzorca w słowie, natomiast niebieskim kolorem zaznaczono wzorzec występujący w podanym tekście.

Wynikiem algorytmu jest ciąg liczb: 0 i 4, które określają pozycje wystąpienia wzorca „KA" w tekście „KARYKATURA".

```
//algorytm realizujacy naiwne wyszukiwanie wzorca w tekscie
#include <iostream>
#include <cstring>
using namespace std;

bool szukaj (string s, string wzorzec, int T[], int &n)
{
 int dlT=s.size(), dlW=wzorzec.size(), j;
```

```
 if (dlW>dlT) return false;
 n=0;
 for (int i=0;i<dlT-dlW+1;i++)
 {
  for (j=i;j<i+dlW;j++)
   if (s[j]!=wzorzec[j-i]) break;
  if (j==i+dlW)
  {
   T[n]=i;
   n++;
  }
 }
 if (n==0) return false;
 return true;
}

int main()
{
 string s, wzorzec;
 int n, T[256]={0};
 cout<<"podaj przeszukiwany tekst: ";
 cin>>s;
 cout<<"podaj wzorzec: ";
 cin>>wzorzec;
 if (szukaj(s,wzorzec,T,n))
 {
  cout<<"liczba wystapien wzorca: "<<n<<endl;
  cout<<"pozycje wystapienia wzorca:"<<endl;
  for (int i=0;i<n;i++) cout<<T[i]<<"\t";
  cout<<endl;
 }
 else cout<<"\nwzorzec nie wystepuje w tekscie"<<endl;
 return 0;
}
```

```
def szukaj(tekst, wzorzec):
  dlT, dlW = len(tekst), len(wzorzec)
  T=[]
  if dlW > dlT:
     return False
  for i in range(dlT - dlW + 1):
    for j in range(i, i+ dlw):
       if tekst[j] != wzorzec[j - i]:
          break
    if j+1 == i+ dlw:
       T.append(i)
  if T == []:
     return False
  return T

 print(szukaj('KARYKATURA','KA'))
 print(szukaj('PARTYTURA', 'KA'))
```

3. **Inne spojrzenie na wyszukiwanie wzorca w tekście** - załóżmy, że zamiast słów rozpatrujemy liczby. W algorytmie omówionym w tym punkcie każdej literze występującej w tekście przypisujemy określoną liczbę.

Przykład

Przeanalizujmy słowo „BBAABBAAB" oraz wzorzec „BAAB". Jeśli znakom występującym w tekście przypiszemy liczby „A" = 0, „B" = 1, to słowo będzie miało postać 110011001, a wzorzec — 1001. W kolejnym kroku algorytmu wybieramy liczbę większą od liczby cyfr we wzorcu, będącą podstawą dalszych obliczeń, na przykład 5. Następnie wyznaczamy wartość, która ma określać wzorzec. Przebieg opisanych działań pokazany jest poniżej.

1	0	0	1
5^3	5^2	5^1	5^0

$$1\cdot 5^3+0\cdot 5^2+0\cdot 5^1+1\cdot 5^0=125+1=126$$

W kolejnym kroku rozpoczynamy wyszukiwanie wzorca w tekście:

1	1	0	0	1	1	0	0	1	
5^3	5^2	5^1	5^0						$= 1\cdot5^3+1\cdot5^2+0\cdot5^1+0\cdot5^0 = 150$
	5^3	**5^2**	**5^1**	**5^0**					**$= 1\cdot5^3+0\cdot5^2+0\cdot5^1+1\cdot5^0 = 126$**
		5^3	5^2	5^1	5^0				$= 0\cdot5^3+0\cdot5^2+1\cdot5^1+1\cdot5^0 = 6$
			5^3	5^2	5^1	5^0			$= 0\cdot5^3+1\cdot5^2+1\cdot5^1+0\cdot5^0 = 30$
				5^3	5^2	5^1	5^0		$= 1\cdot5^3+1\cdot5^2+0\cdot5^1+0\cdot5^0 = 150$
					5^3	**5^2**	**5^1**	**5^0**	**$= 1\cdot5^3+0\cdot5^2+0\cdot5^1+1\cdot5^0 = 126$**

Szarym tłem wyróżniono wzorzec odnaleziony w tekście, a kolorem niebieskim zaznaczono obliczenia, z których wynika, że wzorzec został odnaleziony.

Fragmenty tekstu, które mają wartość równą liczbie przypisanej wzorcowi, najprawdopodobniej są mu równe. Istnieje możliwość pojawienia się tej samej wartości dla dwóch różnych wzorców. W związku z tym w algorytmie należy jeszcze sprawdzić poprawność odnalezionego wystąpienia wzorca.

Zadanie 1

Podaj specyfikację zadania i napisz program wyszukujący w tekście wprowadzonym z klawiatury słowo „napis". Wynikiem wykonania algorytmu powinien być komunikat podający numer pierwszej pozycji, na której dane słowo występuje, lub informacja o tym, że wskazanego słowa w tekście nie znaleziono. Zadanie wykonaj dwoma sposobami: z wykorzystaniem funkcji find() oraz bez zastosowania tej funkcji.

- ✗ `int find(string b) -` wyszukuje w napisie podciąg znaków **b**, zwraca pozycję rozpoczęcia szukanego podciągu w tekście lub wartość większą od długości przeszukiwanego napisu

```
string b="cde";
int a=s.find(b);     // a=2
```

- ✗ **find() –** wyszukuje podciąg i zwraca pozycję pierwszego wystąpienia, zwraca -1 gdy nie znajdzie

```
txt = "Hello, welcome to my world."
x = txt.find("welcome")
print(x)
7
```

Zadanie 2

Podaj specyfikację zadania oraz napisz program, który sprawdzi, czy w tekście wprowadzonym z klawiatury (składającym się ze znaków „A” i „B”) występuje wzorzec „ABAB”. Zastosuj metodę, która każdej literze w tekście przypisuje określoną liczbę.