

Zadanie 1. Kompresja.

Rozważmy algorytm kompresji, który zlicza liczbę kolejnych wystąpień tego samego znaku, a następnie zamiast całej grupy identycznych znaków podaje ten znak tylko jeden raz, poprzedzając go liczbą jego kolejnych wystąpień.

Liczba kolejnych wystąpień każdego znaku nie przekracza 9, więc do zapisania tej liczby wystarczy jeden znak.

Przykład:

tekst źródłowy	tekst skompresowany	rozmiar tekstu w liczbie znaków	
		źródłowego	skompresowanego
FFFYYYYYYYYFFFHAAAAA	3F9Y3F1H5A	21	10

Zadanie 1.1. (2 pkt)

Skompresuj powyższym algorytmem tekst podany w tabeli, oblicz rozmiar tekstu przed kompresją i po kompresji.

tekst źródłowy	tekst skompresowany	rozmiar tekstu w liczbie znaków	
		źródłowego	skompresowanego
***##!!*	3*2#2!1*	8	8

Zadanie 1.2. (1 pkt)

Ile powinna wynosić minimalna liczba kolejnych znaków w grupie, aby jej kompresja była opłacalna?

3

Zadanie 1.3. (1 pkt)

Czy opisana metoda kompresji jest stratna, czy – bezstratna?

bezzatna

Zadanie 1.4. (4 pkt)

Napisz (w postaci listy kroków, schematu blokowego, pseudokodu lub w wybranym języku programowania) algorytm obliczający rozmiar skompresowanego tekstu.

Specyfikacja:

Dane:

n – dodatnia liczba całkowita, długość skompresowanego tekstu

$T[1..n]$ – tablica zawierająca tekst do skompresowania; $T[i]$ – i -ty znak w tekście

Wynik:

b – rozmiar skompresowanego tekstu T

Algorytm:

```
for (int i = 1; i <= n; i++)  
T[i] = ...  
  
int ile = 0;  
char pom = 1;  
string w = "";  
  
for (int i = 2; i <= n; i++)  
{  
    if (T[i] == T[i-1])  
    {  
        ile++;  
    }  
    else  
    {  
        pom = pom + ile + 48;  
        w = w + pom + T[i-1];  
        pom = 0;  
        ile = 1;  
    }  
}  
b = w.length();
```

Wypełnia egzaminator	Nr zadania	1.1.	1.2.	1.3.	1.4.
	Maks. liczba pkt.	2	1	1	4
	Uzyskana liczba pkt.				