Akademia Nauk Stosowanych - Teoretyczne i technologiczne podstawy multimediów - laboratorium			
Temat: LZW – metoda strumieniowej bezstratnej kompresji słownikowej.			Symbol: TiTPM
Nazwisko i imię: Fyda Kamil		Ocena sprawozdania	Zaliczenie:
Data wykonania ćwiczenia: 15.11.2022r.	Grupa: L1		

1. Opis teoretyczny:

LZW (Lempel-Ziv-Welch) – metoda strumieniowej bezstratnej kompresji słownikowej. Najprościej rzecz ujmując algorytm ten buduje pewnego rodzaju słownik wartości, a następnie koduje dane wejściowe za pomocą indeksów elementów tegoż słownika. Algorytm LZW jest dosyć prosty. Wykorzystuje ona fakt, że na obrazie istnieją piksele o powtarzających się kolorach i tworzy na tej podstawie słownik takich powtarzających się ciągów znaków. Całe połączenie zastępowane jest skrótem. Takie przypisanie jest oczywiście zapisywane i dzięki temu dekompresor może odtworzyć plik w postaci nie zmienionej. Bardzo dobrze sprawdza się przy obrazach zawierających duże obszary jednolitego koloru lub tam gdzie istnieje wiele powtarzających się schematów/elementów na obrazie (8-bitowe). Podczas dekompresji czyli otwierania pliku dzieje się sytuacja odwrotna – za pomocą słownika skróty są podmieniane na powtarzające się ciągi znaków. Dzięki temu możemy odtworzyć plik w postaci niezmienionej. Identyczny jak przed kompresją. Jedyny minus jest taki, że słabo radzi sobie z plikami 16-bitowymi. Dzięki LZW można uzyskać plik mniejszy nawet o połowę, jest to więc bardzo zadowalająca metoda kompresji.

2. Opis działania programu:

- Budowanie słownika podstawowego - każdy znak umieszczony będzie w kolejności alfabetycznej.
- Do zmiennej c przypisany zostanie pierwszy znak.
- Dla i od 1 do n-1 gdzie n to długość wiadomości, do s przypisano kolejny znak tej informacji.
- Jeżeli ciąg c + s znajduje się w słowniku do c przypisane zostanie c + s.
- Jeśli c + s nie ma w słowniku to dodajemy c + s do całego słownika, zapisujemy indeks c, następnie do c przypisane zostało s.
- Jeśli i < n to wracamy do punktu 3, w przeciwnym razie zapamiętany zostaje indeks aktualnej wartości c w słowniku.

3. Kod programu:

```cpp
#include <map>
#include <string>
#include <iostream>
#include <vector>
#include <unordered_map>
#include <fstream>
using namespace std;

vector<int> kodowanie(string s1)
{
    cout << "Kodowanie\n";
    unordered_map<string, int> table;
    for (int i = 0; i <= 255; i++) {
        string ch = "";
        ch += char(i);
        table[ch] = i;

    }

    string p = "", c = "";
    p += s1[0];
    int code = 256;
    vector<int> output_code;

    cout << "Sl.podst\tSl.pelny\tKodowanie\n";
    for (int i = 0; i < s1.length(); i++) {

        if (i != s1.length() - 1)
            c += s1[i + 1];
        if (table.find(p + c) != table.end()) {
            p = p + c;

        }
        else {
            cout << p << "\t" << table[p] << "\t\t"
                << p + c << "\t" << code << endl;
            output_code.push_back(table[p]);
            table[p + c] = code;
            code++;

            p = c;
        }

        c = "";

    }
    cout << p << "\t" << table[p] << endl;
    output_code.push_back(table[p]);
    ofstream zapis("wynik.txt");
    for (int i = 0; i < output_code.size(); i++) {
        zapis <<code<<"\t"<<'.'<< p;
    zapis.close(); //zapis pliku - zamkniecie
    }

    return output_code;
}

void dekodowanie(vector<int> op)
{
    cout << "\nDekodowanie\n";
    unordered_map<int, string>table;
    for (int i = 0; i <= 255; i++) {
        string ch = "";
        ch += char(i);
        table[i] = ch;
    }
    int old = op[0], n;
    string s = table[old];
    string c = "";
    c += s[0];
    cout << s;
    int count = 256;
    for (int i = 0; i < op.size() - 1; i++) {
```

```
77          n = op[i + 1];
78          if (table.find(n) == table.end()) {
79              s = table[old];
80              s = s + c;
81          }
82          else {
83              s = table[n];
84          }
85          cout << s;
86          c = "";
87          c += s[0];
88          table[count] = table[old] + c;
89          count++;
90          old = n;
91      }
92  }
93  int main()
94  {
95
96      string s;
97      cout << "Podaj swoj teskst do zakodowania: ";
98      cin >> s;
99      vector<int> output_code = kodowanie(s);
100     cout << "Zakodowana postac: ";
101     for (int i = 0; i < output_code.size(); i++) {
102         cout << output_code[i] << " ";
103     }
104     cout << endl;
105     dekodowanie(output_code);
106
107
108
109     return 0;
110 }
```

4. **Wynik działania programu:**

```
Podaj swoj teskst do zakodowania: wabbawabba
Kodowanie
Sl.podst        Sl.pelny        Kodowanie
w       119             wa      256
a       97              ab      257
b       98              bb      258
b       98              ba      259
a       97              aw      260
wa      256             wab     261
bb      258             bba     262
a       97
Zakodowana postac: 119 97 98 98 97 256 258 97

Dekodowanie
wabbawabba
```

5. **Wynik działania zapisywania do pliku:**