```cpp
/**
*   Teste de compresao para texto
*   Autor Tiago Jordan
*   Github https://github.com/TJordanR
*   Versão 1.0.0
*   TENTAR A SOMA DOS CONJUTOS DE 8 BITS DE DOIS PARES PRA OBTER UM NUMERO INTEIRO DIVIDIDO POR DOIS
*   QUE POR SUA VEZ TERA O VALOR DA SEQUENCIA EXPECIFICA DE CARACTERES REPRESENTANDO OS DOIS PARERS DE
*   CARACTERES CORRESPONDENTE A PALAVRA OU FRASE
**/

#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

using namespace std;

/**
*   Class LerArqBim
*   Define um novo caminho para o arquivo ser lido
*   Ler arquivo
*   Obtem o tamanho do arquivo
*   Salva dados lidos e guarda em uma matriz
**/
class LerArqBim{
private:
    char *Url = "C:\\arquivo.txt";
    char Dados[999999];
    int tamDados = 0;
public:
    /// Define novo caminho do arquivo
    char DefineUrl(char *str){
        Url = str;
    }
    /// Ler o caminho do arquivo e retorna
    char *LerUrlArq(){
        return Url;
    }
    /// Ler os dados do arquivo e salva na variavel
    char LerArqTxt(char *str);
    /// Defini o tamanho da matriz de dados
    int DefineTamDados(int tam){
        tamDados = tam;
    }
    /// Ler o tamanho da matriz de dados
    int LerTamDados(){
        return tamDados;
    }
    /// Ler dados da variavel
    char *AcessaDadosArqMemoria(){
        return Dados;
    }
};

/**
*   Class EncodeDados
*   Ler dados da class e armazena em uma nova matriz para processamento
*   Retorna dados processados
*   Processa dados para uma nova saida em uma matriz
**/
class EncodeDados{
private:
    char *Dados;
    int tamDados = 0;
public:
    /// Recede dados e tamanho da Class LerArqBim
```

```
67      void recebeData(char *str, int tam){
68          Dados = str;
69          tamDados = tam;
70      }
71      /// Retorna dados buff
72      char *strDados(){
73          return Dados;
74      }
75      /// Retorna tamanho do dados
76      int retTamDados(){
77          return tamDados;
78      }
79      /// METODO DE PROCESSAMENTO E CONVERSAO DE DADOS
80      char EncodeBim(char *str);
81  };
82
83  /**
84  *   Class SalveEncode
85  *   Defino um novo caminho para salvar os dados processados
86  *   Retorna o novo caminho definido
87  *   Salva os dados processados no arquivo
88  **/
89  class SalvEncode{
90  public:
91      char *Url_s = "C:\\arquivo.txt";
92      char Dados_s[999999];
93
94      /// Incrementa valor de entrada no contador
95      int contador = -1;
96
97      /// Define novo caminho do arquivo
98      char DefineUrl_s(char *str_s){
99          Url_s = str_s;
100     }
101     /// Ler o caminho do arquivo e retorna
102     char *LerUrlArq_s(){
103         return Url_s;
104     }
105     /// Salva os dados processados do arquivo no arquivo de texto
106     char SalvArqTxt(char *str_s, char *url);
107
108 };
109
110 /**
111 *   Class Utf8bits
112 *   Metodo para definiçao de soma dos bits com base em 16 casas
113 *   Retorna a soma em bits na base 16
114 *   Zera bits para nova contagem
115 *   Seleciona o caracter correspondente e retorna seu valor
116 *   Processa os dados de entrada para conversão dos caracteres
117 *   Salva os dados de saida
118 **/
119 class Utf8bits{
120 private:
121     char alfa16[200] =
"!\"#$%&\()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~?Çüéâäàåçêëèïîì
ÄÅÉæÆôöòûùÿÖÜø£Ø×ƒáíóúñÑªº¿®";
122     int  utf_bit16[16]  = {16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1};
123     int SomaBits16      = 0;
124     int alfa_utf8_num   = 0;
125 public:
126     /// Soma bits da variavel de entra
127     int BistSoma(int s){
128         SomaBits16 = SomaBits16 + s;
129     };
130     /// Retorna o valor da soma
```

```
131     int BitsRetorn(){
132         return SomaBits16;
133     };
134     /// Zera o valor da soma
135     void ZeraBist16(){
136         SomaBits16 = 0;
137     };
138     /// Seleciona a variavel correspondente apartir do valor total da soma
139     char SelectUtf8(int str_c);
140     /// processa os valores de entrada
141     char Processamento(char *str);
142     /// METODO PARA GRAVAR AS SAIDAS NA NOVA MATRIZ
143     SalvEncode *salve   = new SalvEncode;
144     /// Salva os dados no arquivo
145     void SaveDadosArq(){
146         salve->DefineUrl_s("E:\\ENCODE_UTF8_IN_BIM\\saida.txt");
147         cout << salve->LerUrlArq_s() << endl;
148         salve->SalvArqTxt(salve->Dados_s, salve->Url_s);
149     }
150 };
151 
152 /**
153 *   Ler arquivos de entrada e armazena os dados em uma matriz de buff
154 **/
155 char LerArqBim::LerArqTxt(char *str){
156     FILE *arq;
157     arq = fopen(str, "r");
158     if(!arq){
159         printf("Erro nao foi possivel encontrar o caminho ou o arquivo nao existe!\n");
160     }
161     char c;
162     int cont=0;
163     do{
164         c = fgetc(arq);
165         Dados[cont] = c;
166         cont++;
167     }while( c!=EOF );
168     fclose(arq);
169 
170     /// Metodo de gravação do tamanho da matriz de dados str
171     DefineTamDados(cont);
172     printf("Fim do processo de Leitura do arquivo!\n");
173 }
174 
175 /**
176 *   Recebe o valore interiro correspondente a variavel e retorna
177 **/
178 char Utf8bits::SelectUtf8(int str_c){
179 
180     for(int a=0; a<=136; a++){
181         if(a == str_c){
182             alfa_utf8_num = alfa16[a];
183         }
184     }
185     return alfa_utf8_num;
186 }
187 
188 /**
189 *   Processa os dados
190 *   Entrada de duas variaveis para tratamento
191 *   Envia os bits para soma acumulativa dos dados
192 *   Envia dados processados e valor numerico para ser salva em uma arquivo de saida
193 */
194 char Utf8bits::Processamento(char *str){
195 
196     for(int a=0; a<=1; a++){
```

```
        for(int b=0; b<=7; b++){
            if( (str[a] >> b) %2==0 ){
                printf("0");
            }else{
                printf("1");
                BistSoma(utf_bit16[b]);
            }
        }
        printf(" ");
    }

    printf("  Soma bits:%d = ", BitsRetorn());
    printf("%c\n", SelectUtf8(BitsRetorn()));
    /// Contagem dos bits
    salve->contador++;
    /// Envio os dados para salvar
    salve->Dados_s[salve->contador] = SelectUtf8(BitsRetorn());
    /// Zera a variavel de contagem
    ZeraBist16();
}

/**
*   Metodo de leitura da matriz para envio de dados com paredade par para processamente
**/
char EncodeDados::EncodeBim(char *str){

    /// Metodo de auxiliar de processamento dos dados
    Utf8bits *utf8 = new Utf8bits;

    char *str_Dados = strDados();
    int tam = retTamDados();
    for(int a=0; a<=tam-1; a++){
        /// PARIDADE DE "b" = 2 CONVOCA METODO DE PROCESSAMENTO DE DOIS CARACTERES
        if(a%2==1){
            printf("%c %c  ", str_Dados[a-1], str_Dados[a]);
            char ch2[2];
            ch2[0] = str_Dados[a-1];
            ch2[1] = str_Dados[a];
            utf8->Processamento(ch2);
        }
    }
    printf("\nFim do processamento dos dados do arquivo!\n");

    /// Ativa o metodo para salvar os dados no arquivo
    utf8->SaveDadosArq();
}

/**
*   Acessa dados da matriz ja tratada
*   Salva os dados tratados em um arquivo de texto de saida
**/
char SalvEncode::SalvArqTxt(char *str_s, char *url){
    FILE *arq;
    arq = fopen(url, "a");
    if(!arq){
        printf("Erro nao foi possivel encontrar o caminho ou o arquivo nao existe!\n");
    }
    for(int a=0; a<=contador; a++){
        fprintf(arq, "%c", Dados_s[a]);
        printf("%c", Dados_s[a]);
    }
    fclose(arq);
    printf("\nDados gravado com sucesso!\n");
}

int main()
```

```
{
    /// Metodo de definicao do caminho do arquivo
    LerArqBim *darq     = new LerArqBim;

    /// Metodo de leitura do arquivo definido para variavel buff
    LerArqBim *larq     = new LerArqBim;

    /// Metodo de processamento de tratamento de dados
    EncodeDados *endc   = new EncodeDados;

    /// Define um novo caminho para entrada dos dados
    darq->DefineUrl("E:\\ENCODE_UTF8_IN_BIM\\entrada.txt");
    cout << darq->LerUrlArq() << endl;

    /// Ler os dados do arquivo de entrada
    larq->LerArqTxt(darq->LerUrlArq());

    /// Associa os dados para variavel temporaria assim como o tamanho do arquivo
    endc->recebeData(larq->AcessaDadosArqMemoria(), larq->LerTamDados());

    /// Metodo de tratamento
    endc->EncodeBim("Comando");

}
```