

```

/*
 *          File: carattere-utf8.c
 *
 *          Autore: Roberto FULIGNI
 *          Ultima modifica: 04/02/2022
 *
 *          Descrizione: Dato il codepoint di un carattere Unicode,
 *                      stampare a video il carattere e la sua
 *                      codifica UTF-8.
 */

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

#define MAX_BUFFER 7

int main() {
    // In Windows, per stampare correttamente i caratteri codificati
    // in UTF-8, occorre impostare la codepage 65001 (UTF-8)
    SetConsoleOutputCP(CP_UTF8);

    unsigned int codepoint;
    unsigned char buffer[MAX_BUFFER] = {0};          // Tutte le celle sono inizializzate a zero
    printf("Codepoint esadecimale: U+");
    // Lettura da tastiera di un numero esadecimale
    scanf("%X", &codepoint);

    if (codepoint <= 0x7F) {
        // Il codepoint è di 7 bit, si memorizza il suo valore
        // in un solo byte del buffer
        buffer[0] = codepoint & 0b01111111;
    }
    else if (codepoint <= 0x7FF) {
        // Il codepoint è di 11 bit, occorre separare il valore
        // in due gruppi di bit (il primo da 5 bite, il secondo da 6)
        // e memorizzare i gruppi in celle distinte del buffer.
        buffer[0] = 0b11000000 | (codepoint >> 6);          // Primo gruppo
        buffer[1] = 0b10000000 | (codepoint & 0b00111111); // Secondo gruppo
    }
    else if (codepoint <= 0xFFFF) {
        // Codepoint di 16 bit (tre gruppi: 4bit + 6bit + 6bit)
        buffer[0] = 0b11100000 | (codepoint >> 12);          // Primo gruppo
        buffer[1] = 0b10000000 | ((codepoint >> 6) & 0b00111111); // Secondo gruppo
        buffer[2] = 0b10000000 | (codepoint & 0b00111111); // Terzo gruppo
    }
    else if (codepoint <= 0x1FFFFF) {
        // Codepoint di 21 bit (quattro gruppi: 3bit + 6bit + 6bit + 6bit)
        buffer[0] = 0b11110000 | (codepoint >> 18);          // Primo gruppo
        buffer[1] = 0b10000000 | ((codepoint >> 12) & 0b00111111); // Secondo gruppo
        buffer[2] = 0b10000000 | ((codepoint >> 6) & 0b00111111); // Terzo gruppo
        buffer[3] = 0b10000000 | (codepoint & 0b00111111); // Quarto gruppo
    }
    else if (codepoint <= 0x3FFFFFF) {
        // Codepoint di 26 bit (cinque gruppi: 2bit + 6bit + 6bit + 6bit + 6bit)
        buffer[0] = 0b11111000 | (codepoint >> 24);          // Primo gruppo
        buffer[1] = 0b10000000 | ((codepoint >> 18) & 0b00111111); // Secondo gruppo
        buffer[2] = 0b10000000 | ((codepoint >> 12) & 0b00111111); // Terzo gruppo
        buffer[3] = 0b10000000 | ((codepoint >> 6) & 0b00111111); // Quarto gruppo
        buffer[4] = 0b10000000 | (codepoint & 0b00111111); // Quinto gruppo
    }
    else if (codepoint <= 0x7FFFFFFF) {
        // Codepoint di 31 bit (sei gruppi: 1bit + 6bit + 6bit + 6bit + 6bit + 6bit)
        buffer[0] = 0b11111000 | (codepoint >> 30);          // Primo gruppo
        buffer[1] = 0b10000000 | ((codepoint >> 24) & 0b00111111); // Secondo gruppo
    }
}

```

```
    buffer[2] = 0b10000000 | ((codepoint >> 18) & 0b00111111); // Terzo gruppo
    buffer[3] = 0b10000000 | ((codepoint >> 12) & 0b00111111); // Quarto gruppo
    buffer[4] = 0b10000000 | ((codepoint >> 6) & 0b00111111); // Quinto gruppo
    buffer[5] = 0b10000000 | (codepoint & 0b00111111); // Sesto gruppo
}
printf("\n");
printf("    Carattere: %s\n", buffer);
printf("    Codepoint: U+%04X (%u)\n", codepoint, codepoint);
printf("Codifica UTF-8: ");
int i = 0;
// Stampa del contenuto del buffer (dife esadecimali per ogni cella)
// La stampa termina quando si rileva nel buffer il carattere NULL
while (buffer[i] != '\0')
    printf("%02X ", buffer[i++]);
printf("\n\n");

// Premere un tasto per terminare il programma...
system("pause");
return 0;
}
```