```
//*****************************************************************************
//*****************************************************************************
//                                                                            **
//                          Łańcuchy znakowe- konwersje                       **
//                                                                            **
//      funkcjonalności:   konwersje:                                          **
//                         -liczba → łańcuch tekstowy                         **
//                         -łańcuch tekstowy → liczba                         **
//                                                                            **
//*****************************************************************************



#define NIBBLE_bm 0xF

enum Result {OK, ERROR};



void UIntToHexStr (unsigned int uiValue, char pcStr[]){

    unsigned char ucNibbleCounter;
    unsigned char ucCurrentCharacter;

    pcStr[0] = '0';
    pcStr[1] = 'x';

    for(ucNibbleCounter = 0; ucNibbleCounter < 4; ucNibbleCounter ++){

        ucCurrentCharacter = ((uiValue >> (12 - 4 * ucNibbleCounter)) & NIBBLE_bm);

        if(ucCurrentCharacter < 10){
            pcStr[2 + ucNibbleCounter] = ('0' + ucCurrentCharacter);
        }
        else{
            pcStr[2 + ucNibbleCounter] = ('A' + (ucCurrentCharacter-10));
        }
    }
    pcStr[6] = '\0';
}
```

```
enum Result eHexStringToUInt(char pcStr[],unsigned int *puiValue){

    unsigned char ucCharacterCounter;
    unsigned char ucCurrentCharacter;

    *puiValue = 0;

    for(ucCharacterCounter = 2; ucCharacterCounter <= 6; ucCharacterCounter ++){

        if('\0' == pcStr[ucCharacterCounter]){
            return OK;
        }

        ucCurrentCharacter = pcStr[ucCharacterCounter];

        if(('0' <= ucCurrentCharacter) && ('9' >= ucCurrentCharacter)){
            ucCurrentCharacter = ucCurrentCharacter - '0';
        }
        else if(('A' <= ucCurrentCharacter) && ('F' >= ucCurrentCharacter)){
            ucCurrentCharacter = ucCurrentCharacter -'A'+10;
        }
        else{
            return ERROR;
        }
        *puiValue <<= 4;
        *puiValue |= ucCurrentCharacter;
    }
    return ERROR;
}


void AppendUIntToString (unsigned int uiValue, char pcDestinationStr[]){

    unsigned char ucCharacterCounter;

    for(ucCharacterCounter = 0; '\0' != pcDestinationStr[ucCharacterCounter]; ucCharacterCounter ++){}
    UIntToHexStr(uiValue, &pcDestinationStr[ucCharacterCounter]);
}
```