

A.A. 2018/2019 - Elaborato 1

Sostituire tutte le occorrenze di una sottostringa all'interno di una stringa. Ad esempio ("Questa stringa è una stringa", "stringa", "mela") → "Questa mela è una mela".

Input: la stringa in cui effettuare la sostituzione, la stringa da cercare e quella con cui sostituirla (tutti array di BYTE terminanti con 0)

Output: la stringa modificata

Esempi di casi importanti da verificare:

- Sostituzione dell'intera stringa.
- Nuova stringa da sostituire più corta o più lunga.
- Stringa da sostituire con lunghezza 0.

N.B.

- La stringa modificata deve essere terminata da un carattere nullo (byte 0) esattamente come le stringhe di input.
- Si supponga che il buffer di memoria cui corrisponde la stringa da modificare sia sempre sufficientemente ampio per il risultato.

Scheletro da utilizzare per il programma:

```
void main()
{
    // Input
    // La stringa da modificare
    char stringa[1024] = "Questa stringa e' una stringa";
    char str1[] = "stringa";    // la sottostringa da cercare
    char str2[] = "mela";    // la sottostringa con cui sostituirla

    // Output
    char risultato[1024]; // La stringa modificata

    __asm
    {
        // Stampa su video
        printf("Nuova stringa: %s\n", risultato);
    }
```

Attenzione:

- Non cambiare il nome delle variabili del codice C scheletro degli elaborati (attenzione anche alle minuscole/maiuscole).
- Non aggiungere altre variabili C al programma, ma usare solo quelle presenti nel testo degli elaborati.

A.A. 2018/2019 - Elaborato 2

Dato un array di interi senza segno, calcolare il loro M.C.D. (Massimo Comune Divisore).

Scheletro da utilizzare per il programma:

Input: un array di DWORD e il numero di elementi

Output: una DWORD (contenente l'M.C.D.)

Esempi di casi importanti da verificare:

- Vettore contenente uno 0.
- Vettore contenente un 1.
- Vettore con tutti numeri primi.

```
void main()
{
    // Input
    //Array di interi
    unsigned int intArray[] = { 6,21,27 };
    //Il numero di interi nell'array
    int num = sizeof(intArray) / sizeof(intArray[0]);

    // Output
    unsigned int MCD;

    __asm
    {

    }

    // Stampa su video
    printf("M.C.D.: %d \n", MCD);
}
```

Attenzione:

- Non cambiare il nome delle variabili del codice C scheletro degli elaborati (attenzione anche alle minuscole/maiuscole).
- Non aggiungere altre variabili C al programma, ma usare solo quelle presenti nel testo degli elaborati.

A.A. 2018/2019 - Elaborato 3

Dato un array di DWORD, invertire l'ordine dei byte delle DWORD che hanno un numero dispari di bit a uno (l'inversione consiste nel passaggio da little-endian a big-endian).

Scheletro da utilizzare per il programma:

Input: un array di DWORD e il numero di elementi

Output: lo stesso array, modificato come richiesto

N.B.

- Per controllare se il risultato del programma è corretto può essere utile visualizzare gli elementi dell'array in base 2.

```
void main()
{
    // Vettore di DWORD da modificare
    unsigned int vet[] = { 0,1,2,3,4,5,6,7,8,9,10,-1,-2,-3,254 };
    // Numero di elementi del vettore
    unsigned int num = sizeof(vet) / sizeof(vet[0]);

    // Blocco assembler
    __asm
    {
        // Stampa su video
        {
            unsigned int i;
            for (i = 0; i < num; i++)
                printf("0x%08X,", vet[i]);
        }
    }
}
```

Attenzione:

- Non cambiare il nome delle variabili del codice C scheletro degli elaborati (attenzione anche alle minuscole/maiuscole).
- Non aggiungere altre variabili C al programma, ma usare solo quelle presenti nel testo degli elaborati.