A.A. 2016/2017 - Elaborato 1

Data una stringa contenente data e ora corrente nel formato gg/mm/aaaa hh:mm:ss, scomporla in giorno, mese, anno, ora, minuto e secondo.

Input: una stringa (contenente la data nel formato gg/mm/aa hh:mm:ss), sotto forma di array di BYTE terminante con 0.

Output: 6 WORD (contenenti giorno, mese, anno, ora, minuto e secondo).

Scheletro da utilizzare per il programma:

```
#include <stdio.h>


void main()
{
    // Input
    char data[] = "20/03/1979 12:23:44";    //gg/mm/aaaa hh:mm:ss

    // Output
    unsigned short giorno;
    unsigned short mese;
    unsigned short anno;
    unsigned short ora;
    unsigned short minuto;
    unsigned short secondo;

    __asm
    {
    }

    // Stampa su video
    printf("Giorno: %i Mese: %i Anno: %i Ora: %i Minuto: %i Secondo: %i \n",
            giorno,mese,anno,ora,minuto,secondo);
}
```

A.A. 2016/2017 - Elaborato 2

Fornito un insieme di punti bidimensionali nel piano cartesiano, trovare il punto più vicino e il punto più lontano da un punto (x,y) dato. *Ogni punto è rappresentato da una DWORD: la WORD meno significativa è la coordinata x, la più significativa la y.*

Input: un array di DWORD (l'insieme dei punti), una DWORD (numero di punti nell'insieme) e una DWORD (il punto x,y).

Output: due DWORD (indice nell'array del punto più vicino e del punto più lontano).

Esempi di casi importanti da verificare:

- Insieme contenente un solo punto.
- Alcuni punti con una o entrambe le coordinate negative.

N.B.

- Le coordinate dei punti (che assumono solo valori interi) possono essere sia positive che negative, comprese nell'intervallo [-1000,1000].
- Nel caso di punti equidistanti dovrà essere restituito l'indice con valore minore.

Scheletro da utilizzare per il programma:

```
#include <stdio.h>


void main()
{
    // Input
    unsigned int Point = 0xFFFFFFFF;
    unsigned int PointSet[] = { (10<<16) | 3, (4<<16) | 2, 0xFFFF0002 };    // insieme
    unsigned int n = sizeof(PointSet)/sizeof(PointSet[0]);  // numero punti nell'insieme

    // Output
    unsigned int idxVicino;     // risultato: indice del punto piu' vicino a Point
    unsigned int idxLontano;    // risultato: indice del punto piu' vicino a Point

    __asm
    {
    }

    // Stampa su video
    printf("Il punto piu' vicino a (%d,%d) e' (%d,%d) [indice=%d]\n",
        (short int)(Point&0xFFFF),(short int)(Point>>16),
        (short int)(PointSet[idxVicino]&0xFFFF),(short int)(PointSet[idxVicino]>>16),
        idxVicino);
    printf("Il punto piu' lontano a (%d,%d) e' (%d,%d) [indice=%d]\n",
        (short int)(Point&0xFFFF),(short int)(Point>>16),
        (short int)(PointSet[idxLontano]&0xFFFF),(short int)(PointSet[idxLontano]>>16),
        idxLontano);

}
```

A.A. 2016/2017 - Elaborato 3

Data una sequenza di bit, contare il numero di transizioni 0→1 e 1→0.

Input: un array di BYTE da considerare come una sequenza di bit; una WORD (il numero totale di bit).

Output: due DWORD (il numero totale di transizioni 0→1 e 1→0).

N.B.

- viene specificata in input la lunghezza in *bit*: può quindi accadere che l'ultimo BYTE dell'array sia utilizzato solo in parte.
- I *bit* all'interno di ciascun BYTE devono essere analizzati dal meno significativo al più significativo.

Scheletro da utilizzare per il programma:

```
#include <stdio.h>


void main()
{
    // Input
    unsigned char vet[]={0xAA,0xFC,0x09};   //Array di byte (da considerare come
                                            //una sequenza di bit)
    unsigned short int len=19;              //Lunghezza (numero di bit)

    // Output
    unsigned int transizioni01;             //Numero di transizioni 0->1
    unsigned int transizioni10;             //Numero di transizioni 1->0

    __asm
    {
    }

    // Stampa su video
    printf("Le transizioni 0->1 presenti sono: %d\n",transizioni01);
    printf("Le transizioni 1->0 presenti sono: %d\n",transizioni10);
}
```