

23/1/2019

ftp://ftp.elet.polimi.it/users/ALESSANDRO.GABRIELLI/Didattica/Fondamenti_di_informatica/2018_2019/laboratori/Lab4/codice_lab04/lab...

```

#include <stdio.h>
#define CAPIENZA 4
#define TERMINATORE '-'

typedef struct
{
    char Dati[CAPIENZA];
    int Utili;
} TipoL;
/* tipo lista sequenziale utile per contenere una sequenza di char */

typedef enum {false, true} boolean;

void Stampa(TipoL ListaDaStampare);
/* riceve una lista sequenziale di char e ne stampa il contenuto */

char Minimo(TipoL ListaDaEsaminare);
/* riceve una lista sequenziale di char contenente caratteri minuscoli e restituisce il valore minimo
in essa contenuto, ovvero il carattere piu' vicino all'inizio dell'alfabeto. La funzione opera
correttamente solo se la lista contiene almeno un carattere.*/

void EseguiInput(TipoL *ListaDaRiempire);
/* riceve una lista sequenziale di char e inserisce in essa i dati richiedendoli all'utente */

int main()
{
    TipoL Lista; /* lista sequenziale di interi inseriti dall'utente */

    /** FASE DI INPUT ***/
    EseguiInput(&Lista);

    /** FASE DI STAMPA ***/
    Stampa(Lista);

    /** FASE DI STAMPA DEL MINIMO ***/
    if (Lista.Utili > 0)
    {
        printf("\nMinimo: %c\n", Minimo(Lista));
    }

    return (0);
}

void Stampa(TipoL ListaDaStampare)
{
    int ContaStampati;
    /* numero di elementi della lista stampati a schermo */

    printf("\n\nContenuto della lista:\n{ ");
    for (ContaStampati = 0; ContaStampati < ListaDaStampare.Utili; ++ContaStampati)
    {
        printf("%c ", ListaDaStampare.Dati[ContaStampati]);
    }
    printf("}\n");
}

char Minimo(TipoL ListaDaEsaminare)
{
    char Min;
    /* elemento di minimo valore finora trovato */
    int ContaElem;
    /* usato per scorrere il campo Dati */

    Min = ListaDaEsaminare.Dati[0];

```

```

23/1/2019      ftp://ftp.elet.polimi.it/users/ALESSANDRO.GABRIELLI/Didattica/Fondamenti_di_informatica/2018_2019/laboratori/Lab4/codice_lab04/lab...
/* Nota: la lista contiene sempre almeno un elemento */

for (ContaElem = 1; ContaElem < ListaDaEsaminare.Utili; ++ContaElem)
{
    if (ListaDaEsaminare.Dati[ContaElem] < Min)
    {
        Min = ListaDaEsaminare.Dati[ContaElem];
    }
}

return (Min);
}

void EseguiInput(TipoL *ListaDaRiempire)
{
    boolean FineInput;
    /* controlla la fine della fase di input */
    int ContaElementi;
    /* conta gli elementi della lista gia' esaminati */
    boolean GiaPresente;
    /* vale true se nuovo elemento e' un duplicato di un elemento
     * gia' presente nella lista */

    ListaDaRiempire->Utili = 0;
    /* svuota la lista */

    printf("\n\nInserisci da 0 a %d caratteri minuscoli diversi tra loro.", CAPIENZA);
    printf("\n[Il carattere '%c' termina l'input. Ogni altro carattere sara' ignorato.]\n",
    TERMINATORE);

    FineInput = false;

    while ((false == FineInput) && ListaDaRiempire->Utili < CAPIENZA)
    {
        scanf("%c", &ListaDaRiempire->Dati[ListaDaRiempire->Utili]);
        scanf("%*c");
        /* elimina il carattere '\n' inserito dall'utente premendo <enter> e che
         * altrimenti verrebbe letto alla prossima iterazione del ciclo */

        if ( (ListaDaRiempire->Dati[ListaDaRiempire->Utili] >= 'a') && (ListaDaRiempire-
>Dati[ListaDaRiempire->Utili] <= 'z') )
        /* se il valore inserito e' utile */
        {
            /* verifica se il valore inserito era gia' presente */
            ContaElementi = 0;
            GiaPresente = false;

            while (ContaElementi < ListaDaRiempire->Utili)
            {
                /* printf("\nDEBUG: sto esaminando l'elemento di indice %d \n", ContaElementi); */

                if (ListaDaRiempire->Dati[ContaElementi] == ListaDaRiempire->Dati[ListaDaRiempire->Utili])
                /* se l'elemento in esame e' identico a quello che si vuole
                 * inserire nella lista sequenziale */
                {
                    GiaPresente = true;
                }
                ++ContaElementi;
            }

            if (false == GiaPresente)
            {
                ++ListaDaRiempire->Utili;
                /* solo se ListaDaRiempire->Utili viene incrementato si completa

```

```
23/1/2019      ftp://ftp.elet.polimi.it/users/ALESSANDRO.GABRIELLI/Didattica/Fondamenti_di_informatica/2018_2019/laboratori/Lab4/codice_lab04/lab...
                * effettivamente l'inserimento del nuovo elemento */
            }
            else
            {
                printf("Il carattere '%c' e' un doppione ed e' stato ignorato.\n", ListaDaRiempire-
>Dati[ListaDaRiempire->Utili]);
            }
        }
        else
        {
            if ( TERMINATORE == ListaDaRiempire->Dati[ListaDaRiempire->Utili] )
            {
                FineInput = true;
            }
            else
            {
                printf("Il carattere '%c' non e' valido ed e' stato ignorato.\n", ListaDaRiempire-
>Dati[ListaDaRiempire->Utili]);
            }
        }
    }
}
```