

1

Presentiamo due possibili soluzioni che si differenziano per il modo con cui i risultati vengono comunicati dalla funzione al chiamante. Nella prima versione (`conta_pari`) i due parametri vengono passati per indirizzo; nella seconda (`conta_pari2`) la funzione è di tipo `int` e ritorna il numero di pari, mentre il numero dei dispari è ancora restituito per indirizzo.

```

struct elemento {
    int inf;
    struct elemento *pun;
};

struct elemento *crea_lista2();
void visualizza_lista(struct elemento *);
void conta_pari(struct elemento *, int *, int *);
int conta_pari2(struct elemento *, int *);

main()
{
    int pari, dispari;
    struct elemento *punt_lista;

    punt_lista = crea_lista2();
    visualizza_lista(punt_lista);

    /* chiamata prima versione di conta pari */
    conta_pari(punt_lista, &pari, &dispari);
    printf("\nPari: %d    Dispari: %d", pari, dispari);

    /* chiamate seconda versione di conta pari */
    printf("\nPari: %d", conta_pari2(punt_lista, &dispari));
    printf("    Dispari: %d\n", dispari);
}

void conta_pari(struct elemento *p, int *ppari, int *pdispari)
{
    *ppari = *pdispari = 0;

    while(p!=NULL) {
        if(p->inf % 2 == 0)
            (*ppari)++;
        else
            (*pdispari)++;
        p = p->pun;
    }
}

conta_pari2(struct elemento *p, int *pdispari)
{
    int pari = 0;
    *pdispari = 0;

    while(p!=NULL) {
        if(p->inf % 2 ==0)
            pari++;
        else
            (*pdispari)++;
        p = p->pun;
    }
    return(pari);
}

```