

# *Soluzioni*

## **Esercizio 1:**

Scrivere una funzione ricorsiva che riceva un intero e stampi ricorsivamente un numero corrispondente di punti '.'

## **Soluzione:**

```
void stampa_punti(int n) {
    if(n==0) return;

    stampa_punti(n-1);

    printf(".");
}
```

## **Esercizio 6:**

Scrivere una funzione ricorsiva che, dato un array di interi, inverta le posizioni dei suoi elementi.

Ad esempio l'invocazione `inverti(a)`, dove `a` e' l'array `{1, 2, 3, 4}`, deve modificare l'array in `{4, 3, 2, 1}`.

## **Soluzione:**

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define LUNG 20

void inverti(int vet[], int dim);

main() {
    int vett[LUNG], i, n;

    srand(time(NULL));

    printf("Inserire la lunghezza dell'array (max %d): ", LUNG);
    i=scanf("%d", &n);
    if(i==0 || n>LUNG) return;

    for(i=0; i<n; i++) {
        vett[i]=rand()%100;
        printf("%d ", vett[i]);
    }
    printf("\n");

    inverti(vett, n);
}
```

```

    for(i=0; i<n; i++)
        printf("%d ", vett[i]);
    printf("\n");
}

/*
*/
void inverti(int vet[], int dim) {
    int temp;

    if(dim<2) return;

    /* scambiamo il primo e l'ultimo elemento dell'array
    */
    temp=vet[0];
    vet[0]=vet[dim-1];
    vet[dim-1]=temp;

    /* ricorriamo sull'array escluso il primo e l'ultimo elemento */
    inverti(vet+1, dim-2);
}

```

### **Esercizio 7:**

Sulla base dell'esercizio 1, riuscite a far stampare ricorsivamente punti '.' alternati a '\_'? E punti '.' alternati a '\_' e a '='? Qual'e' il metodo generale?

Esempio di interazione col programma

Inserire un numero: 7

7 -> .\_.\_.\_.

Inserire un numero: 9

9 -> .\_.=.\_.=.

### **Soluzione:**

Notate che la stampa deve sempre partire con un punto: questo vi costringe a una fra queste due soluzioni:

- ricorrere sempre sulla chiusura, invece che sull'apertura, altrimenti non siete in grado di dire quando siete all'inizio;

- usare una funzione ausiliaria che chiami lo worker (la funziona che fa il lavoro vero e proprio) con un parametro in piu', inizialmente settato a 0.

Nel primo caso abbiamo una funzione sola ma essendo invertita non pu' stampare l'invio a capo, che deve essere stampato dal main.

Nel secondo caso la funzione fa tutto ma ha bisogno di una seconda funzione ausiliaria che la invochi nel modo giusto (passando il secondo parametro a 0).

Qual'e' il metodo generale? L'unica cosa che potete fare, in una funzione ricorsiva, per distinguere le varie invocazioni e' contare: finche' c'e' un ordine chiaro, la funzione ricorsiva puo' stampare la stringa senza problemi. In questo esercizio, quindi, distinguiamo fra pari e dispari e, nei pari, fra i

multipli di 4 e no.

```
void stampa_punti(int n);
void stampa_punti_linee(int n);
void stampa_punti_linee2(int n);
void stampa_punti_linee_uguale(int n);
/*
  Un main semplice che chiede un valore all'utente e lo passa alle funzioni
*/
main() {
  int a,i;

  printf("Inserire un intero: ");
  i=scanf("%d", &a);
  if(i==0) return;

  stampa_punti(a);
  printf("\n");

  stampa_punti_linee(a);
  printf("\n");

  stampa_punti_linee_uguale(a);
  printf("\n");

  stampa_punti_linee2(a);
}

void stampa_punti(int n) {
  if(n==0) return;

  stampa_punti(n-1);

  printf(".");
}

void stampa_punti_linee(int n){
  if(n==0) return;

  stampa_punti_linee(n-1);

  if(n%2==0) printf("_");
  else printf(".");
}

void stampa_punti_linee_worker(int a, int n) {
  if(a==n) {
    printf("\n");
    return;
  }

  if(a%2==0) printf(".");
  else printf("_");
}
```

```

    stampa_punti_linee_worker(a+1,n);
}

void stampa_punti_linee2(int n) {
    stampa_punti_linee_worker(0,n);
}

void stampa_punti_linee_uguale(int n){
    if(n==0) return;

    stampa_punti_linee_uguale(n-1);

    if(n%2==0) {
        if(n%4==0) printf("=");
        else printf("_");
    }
    else printf(".");
}

```

### **Esercizio 8:**

Scrivere una funzione iterativa e ricorsiva che presi in input due vettori A e B lunghi n stampi insieme il primo valore di A e l'ultimo valore di B, poi il secondo valore di A e il penultimo di B e così via.

### **Soluzione:**

```

#include <stdio.h>
void stampaVettori(int A[], int B[], int n, int i) {
    if (i >= n) {
        return;
    }
    printf("%d - %d\n", A[i], B[n-1-i]);
    stampaVettori(A, B, n, i+1);
}

int main () {
    leggi Input;
    stampaVettori(A, B, n, 0);
    return 0;
}

#include <stdio.h>
void stampaVettori(int A[], int B[], int n) {
    if (n <= 0) {
        return;
    }
    printf("%d - %d\n", *A, *(B + n - 1));
    stampaVettori(A+1, B, n-1);
}

int main () {
    leggi Input;
    stampaVettori(A, B, n);
    return 0;
}

```

**Esercizio 10:**

Dati due vettori di interi A e B di lunghezza n, si ottenga di scambiare il primo elemento del vettore A con l'ultimo del vettore B, il secondo di A con il penultimo di B e così via. Per scambiare gli elementi si faccia uso di una funzione a cui siano passati solo gli elementi da scambiare e non l'intero vettore. Definire una funzione scambia. Richiamare la funzione scambia sugli opportuni valori dei vettori

**Soluzione:**

```
#include <stdio.h>
void scambia(int *a, int *b) {
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    leggi A, B, n;
    scambiaVettori(A, B, n, 0);
    return 0;
}

void ScambiaVettori (int A[], int B[], int n, int i) {
    if (i >= n) return;
    scambia(&A[i], &B[n-1-i]);
    scambiaVettori(A, B, n, i+1);
}
```