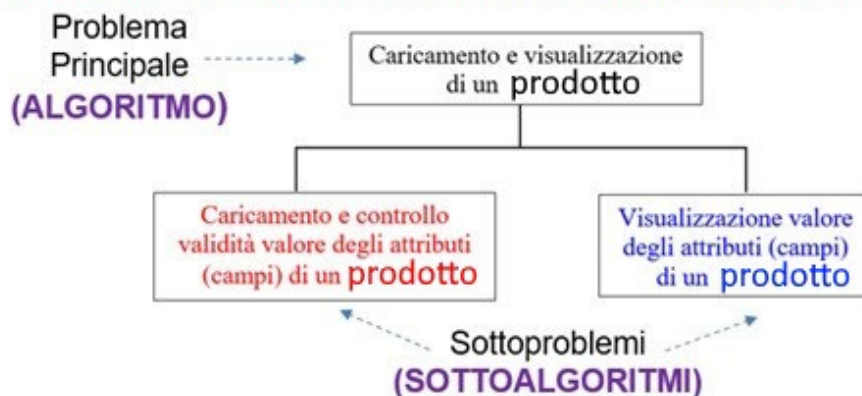


PROBLEMA: Caricamento e visualizzazione di un record di tipo "Prodotto" il cui tracciato è assegnato secondo la seguente rappresentazione tabellare (SOTTOPROGRAMMI):

Numero	Nome Campo	Tipo Campo	Lunghezza	Descrizione	FlagSubRec
1	CodProdotto	ARRAY DI CHAR	6	Codice del Prodotto (obbligatorio)	
2	NomeProdotto	ARRAY DI CHAR	25	Nome del prodotto (obbligatorio)	
3	Giacenza	INT	4	Giacenza del prodotto	
4	Prezzo	REAL	3,2	Prezzo unitario del prodotto	

N.B. Utilizzando la la **METODOLOGIA di PROGETTAZIONE TOP-DOWN** il problema assegnato potrebbe essere scomposto nei seguenti sottoproblemi:



Grazie alla metodologia di progettazione top-down, abbiamo dunque individuato seguenti **SOTTOPROGRAMMI** (procedura o funzione) da implementare, che si devono occupare delle seguenti azioni:

- 1) il **caricamento** (con eventuale controllo del valore di tutti gli attributi (i campi))
- 2) la **visualizzazione** di tutti gli attributi (campi) di un **RECORD** con il seguente **tracciato (PSEUDOCODIFICA)**:

TIPO Prodotto = RECORD

CodProdotto : **ARRAY [6] DI CHAR**
 NomeProdotto : **ARRAY [25] DI CHAR**
 Giacenza : **INT**
 Prezzo : **REAL**

FINE RECORD

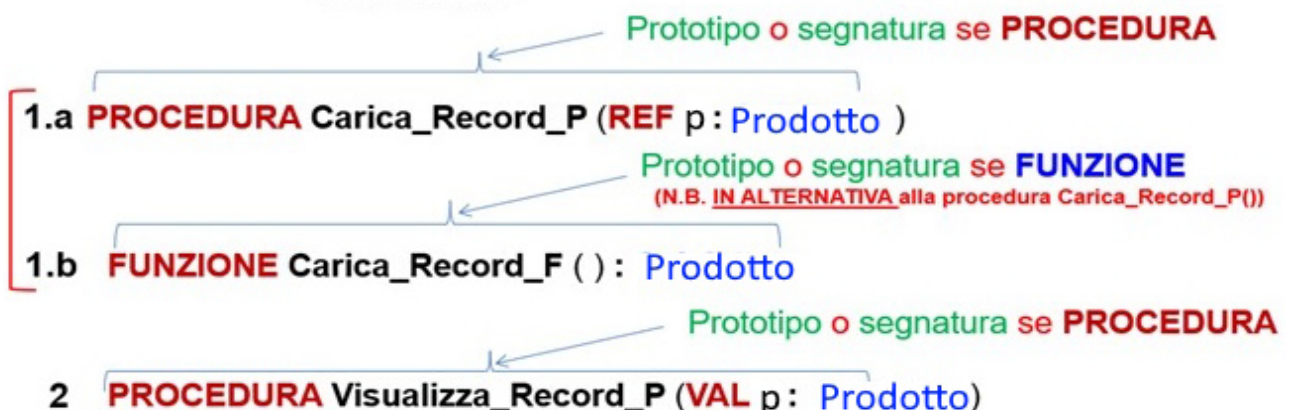


TABELLE DEI DATI

DATI DI INPUT DEL PROBLEMA PRINCIPALE PROCEDURA main()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

DATI DI OUTPUT DEL PROBLEMA PRINCIPALE PROCEDURA main()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

DATI DI ELABORAZIONE o DI LAVORO DEL PROBLEMA PRINCIPALE PROCEDURA main()				
Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
p	Prodotto	STATICA	Vedi TRACCIATO record	Istanza generica del tipo Prodotto
MAXCHARCODICE	INT	STATICA	6	Massimo numero di caratteri per Codice Prodotto
MAXCHARNOME	INT	STATICA	30	Massimo numero di caratteri per Nome Prodotto

ALGORITMO Record_E_Sottoprogrammi

MAXCHARCODICE 6
 MAXCHARNOME 25

TIPO Prodotto = RECORD

CodProdotto : **ARRAY**[MAXCHARCODICE] **DI CHAR**
 NomeProdotto : **ARRAY**[MAXCHARNOME] **DI CHAR**
 Giacenza : **INT**
 Prezzo : **REAL**

FINE RECORD

PROCEDURA main()

p : **Prodotto**

INIZIO

//Utilizziamo la procedura 1.a

Carica_Record_P (p)

//Oppure potremmo utilizzare IN ALTERNATIVA la funzione 1.b

//p ← Carica_Record_F()

//Utilizziamo la procedura 2

Visualizza_Record_P (p)

RITORNA

FINE

TABELLE DEI DATI

DATI DI INPUT DEL SOTTOPROBLEMA: PROCEDURA Carica_Record_P()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
p	Prodotto	STATICA	Vedi TRACCIATO record	Parametro passato per RIFERIMENTO

DATI DI OUTPUT DEL SOTTOPROBLEMA: PROCEDURA Carica_Record_P()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
p	Prodotto	STATICA	Vedi TRACCIATO record	Parametro passato per RIFERIMENTO

DATI DI ELABORAZIONE o DI LAVORO DEL SOTTOPROBLEMA: PROCEDURA Carica_Record_P()				
Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

PROCEDURA Carica_Record_P (**REF** p : Prodotto)

INIZIO

RIPETI

Scrivi("Codice prodotto = ")

 Leggi (p.Codice)

FINCHE' (Lunghezza(p.CodProdotto) ≠ 0) **AND**

 (Lunghezza(p. CodProdotto) ≤ MAXCHARCODICE)

RIPETI

Scrivi("Nome prodotto = ")

 Leggi (p.NomeProdotto)

FINCHE' (Lunghezza(p. NomeProdotto) ≠ 0) **AND**

 (Lunghezza(p. NomeProdotto) ≤ MAXCHARNOME)

RIPETI

Scrivi("Giacenza prodotto = ")

 Leggi (p.Giacenza)

FINCHE' (p.Giacenza ≥ 0) **AND** (p.Giacenza ≤ 9999)

RIPETI

Scrivi("Prezzo prodotto = ")

 Leggi (p.Prezzo)

FINCHE' (p.Prezzo ≥ 0) **AND** (p.Prezzo ≤ 9.99)

RITORNA

FINE

ALTERNATIVA: TABELLE DEI DATI

DATI DI INPUT DEL SOTTOPROBLEMA: FUNZIONE Carica_Record_F()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

DATI DI OUTPUT DEL SOTTOPROBLEMA: FUNZIONE Carica_Record_F()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

DATI DI ELABORAZIONE o DI LAVORO DEL SOTTOPROBLEMA: FUNZIONE Carica_Record_F()				
Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
p	Prodotto	STATICA	Vedi TRACCIATO record	Prodotto generico restituito nel nome della funzione

FUNZIONE Carica_Record_F () : Prodotto

p : Prodotto

INIZIO

RIPETI

Scrivi("Codice prodotto = ")

Leggi (p.Codice)

FINCHE' (Lunghezza(p.CodProdotto) \neq 0) **AND**
(Lunghezza(p. CodProdotto) \leq MAXCHARCODICE)

RIPETI

Scrivi("Nome prodotto = ")

Leggi (p.NomeProdotto)

FINCHE' (Lunghezza(p. NomeProdotto) \neq 0) **AND**
(Lunghezza(p. NomeProdotto) \leq MAXCHARNOME)

RIPETI

Scrivi("Giacenza prodotto = ")

Leggi (p.Giacenza)

FINCHE' (p.Giacenza \geq 0) **AND** (p.Giacenza \leq 9999)

RIPETI

Scrivi("Prezzo prodotto = ")

Leggi (p.Prezzo)

FINCHE' (p.Prezzo \geq 0) **AND** (p.Prezzo \leq 9.99)

RITORNA (p)

FINE

TABELLE DEI DATI

DATI DI INPUT DEL SOTTOPROBLEMA: PROCEDURA Visualizza_Record_P()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
p	Prodotto	STATICA	Vedi TRACCIATO record	Parametro passato per VALORE

DATI DI OUTPUT DEL SOTTOPROBLEMA: PROCEDURA Visualizza_Record_P()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
p	Prodotto	STATICA	Vedi TRACCIATO record	Parametro passato per VALORE

DATI DI ELABORAZIONE o DI LAVORO DEL SOTTOPROBLEMA: PROCEDURA Visualizza_Record_P()				
Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

PROCEDURA Visualizza_Record_P (**VAL** p : **Prodotto**)

INIZIO

Scrivi("Codice prodotto: ")

Scrivi (p.CodProdotto))

Scrivi("Nome prodotto: ")

Scrivi (p.NomeProdotto)

Scrivi("Giacenza prodotto: ")

Scrivi (p.Giacenza)

Scrivi("Prezzo prodotto: ")

Scrivi (p.Prezzo)

RITORNA

FINE

ALGORITMO Record_E_Sottoprogrammi

MAXCHARCODICE 6
MAXCHARNOME 25

TIPO Prodotto = RECORD

CodProdotto : **ARRAY**[MAXCHARCODICE] **DI CHAR**

NomeProdotto : **ARRAY**[MAXCHARNOME] **DI CHAR**

Giacenza : **INT**

Prezzo : **REAL**

FINE RECORD

PROCEDURA main()

p : **Prodotto**

INIZIO

//Utilizziamo la procedura 1.a

Carica_Record_P (p)

//Oppure potremmo utilizzare **IN ALTERNATIVA** la funzione 1.b

//p ← **Carica_Record_F**()

//Utilizziamo la procedura 2

Visualizza_Record_P (p)

RITORNA

FINE

RIPETI

Scrivi("Codice prodotto = ")

 Leggi (p.Codice)

FINCHE' (Lunghezza(p.CodProdotto) ≠ 0) **AND**
 (Lunghezza(p. CodProdotto) ≤ MAXCHARCODICE)

RIPETI

Scrivi("Nome prodotto = ")

 Leggi (p.NomeProdotto)

FINCHE' (Lunghezza(p. NomeProdotto) ≠ 0) **AND**
 (Lunghezza(p. NomeProdotto) ≤ MAXCHARNOME)

RIPETI

Scrivi("Giacenza prodotto = ")

 Leggi (p.Giacenza)

FINCHE' (p.Giacenza ≥ 0) **AND** (p.Giacenza ≤ 9999)

RIPETI

Scrivi("Prezzo prodotto = ")

 Leggi (p.Prezzo)

FINCHE' (p.Prezzo ≥ 0) **AND** (p.Prezzo ≤ 9.99)

RITORNA

FINE

FUNZIONE Carica_Record_F () : **Prodotto**

p : Prodotto

INIZIO

RIPETI

Scrivi("Codice prodotto = ")

Leggi (p.Codice)

FINCHE' (Lunghezza(p.CodProdotto) ≠ 0) **AND**
(Lunghezza(p. CodProdotto) ≤ MAXCHARCODICE)

RIPETI

Scrivi("Nome prodotto = ")

Leggi (p.NomeProdotto)

FINCHE' (Lunghezza(p. NomeProdotto) ≠ 0) **AND**
(Lunghezza(p. NomeProdotto) ≤ MAXCHARNOME)

RIPETI

Scrivi("Giacenza prodotto = ")

Leggi (p.Giacenza)

FINCHE' (p.Giacenza ≥ 0) **AND** (p.Giacenza ≤ 9999)

RIPETI

Scrivi("Prezzo prodotto = ")

Leggi (p.Prezzo)

FINCHE' (p.Prezzo ≥ 0) **AND** (p.Prezzo ≤ 9.99)

RITORNA (p)

FINE

PROCEDURA Visualizza_Record_P (**VAL** p : **Prodotto**)

INIZIO

Scrivi("Codice prodotto: ")

Scrivi (p.CodProdotto))

Scrivi("Nome prodotto: ")

Scrivi (p.NomeProdotto)

Scrivi("Giacenza prodotto: ")

Scrivi (p.Giacenza)

Scrivi("Prezzo prodotto: ")

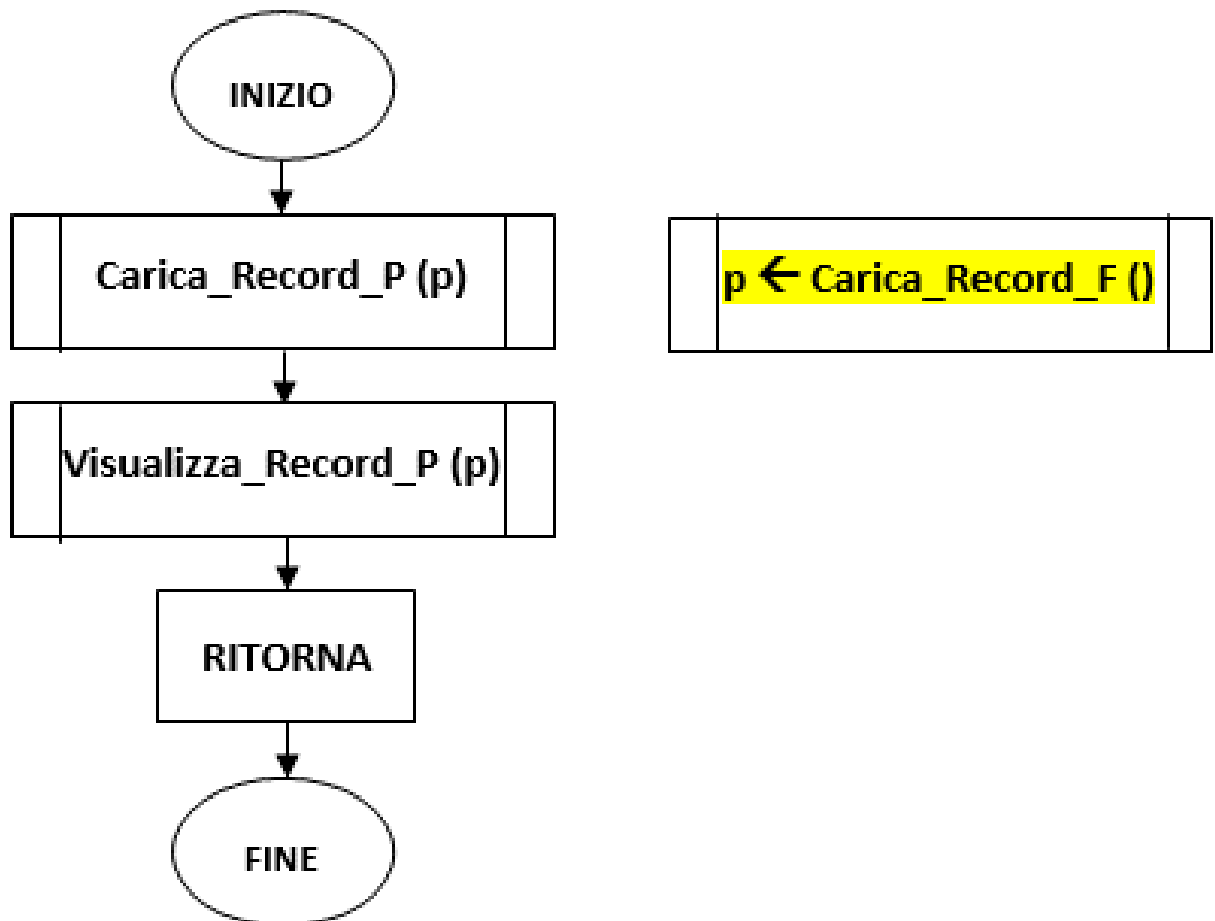
Scrivi (p.Prezzo)

RITORNA

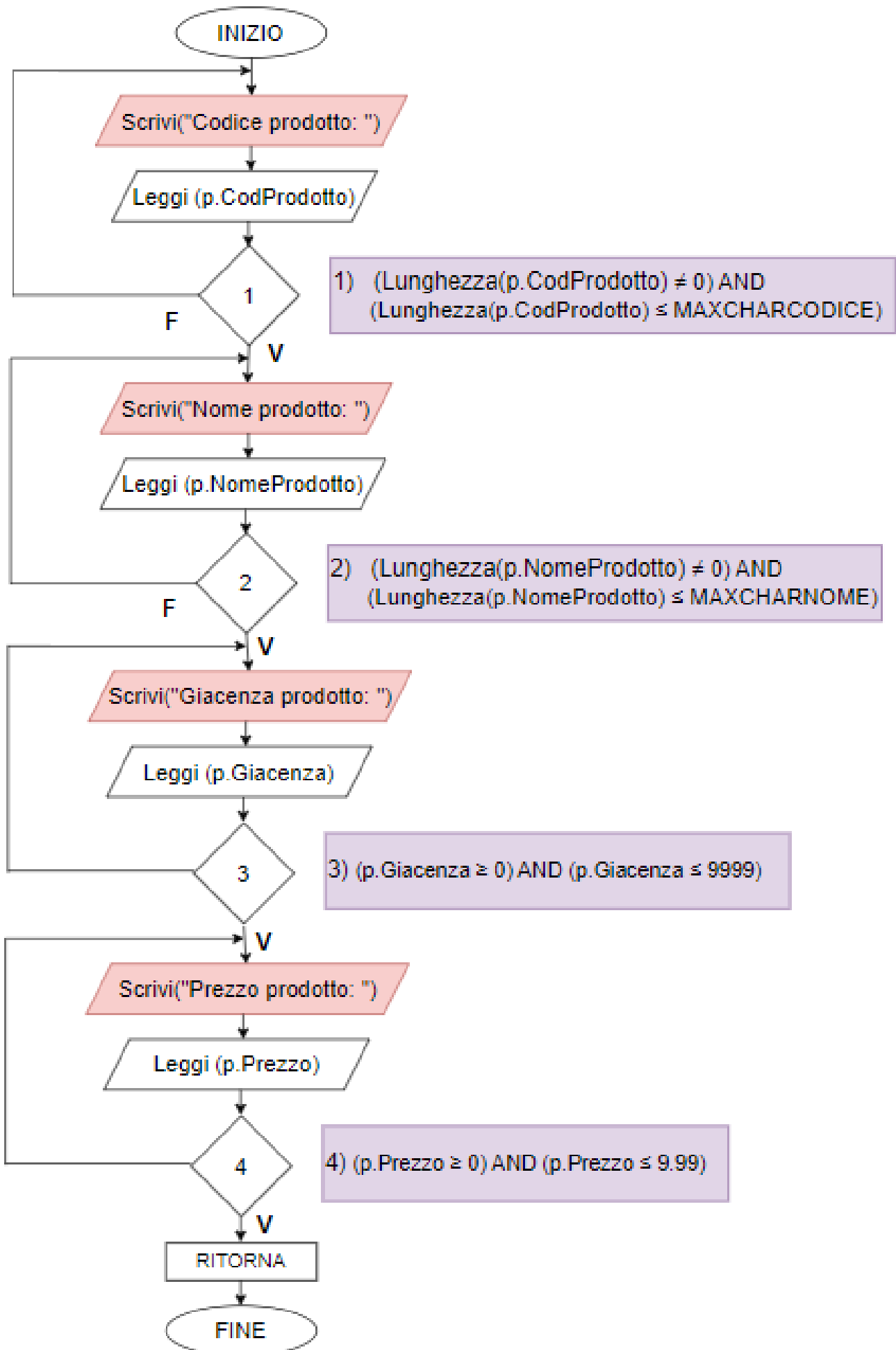
FINE

FLOWCHART

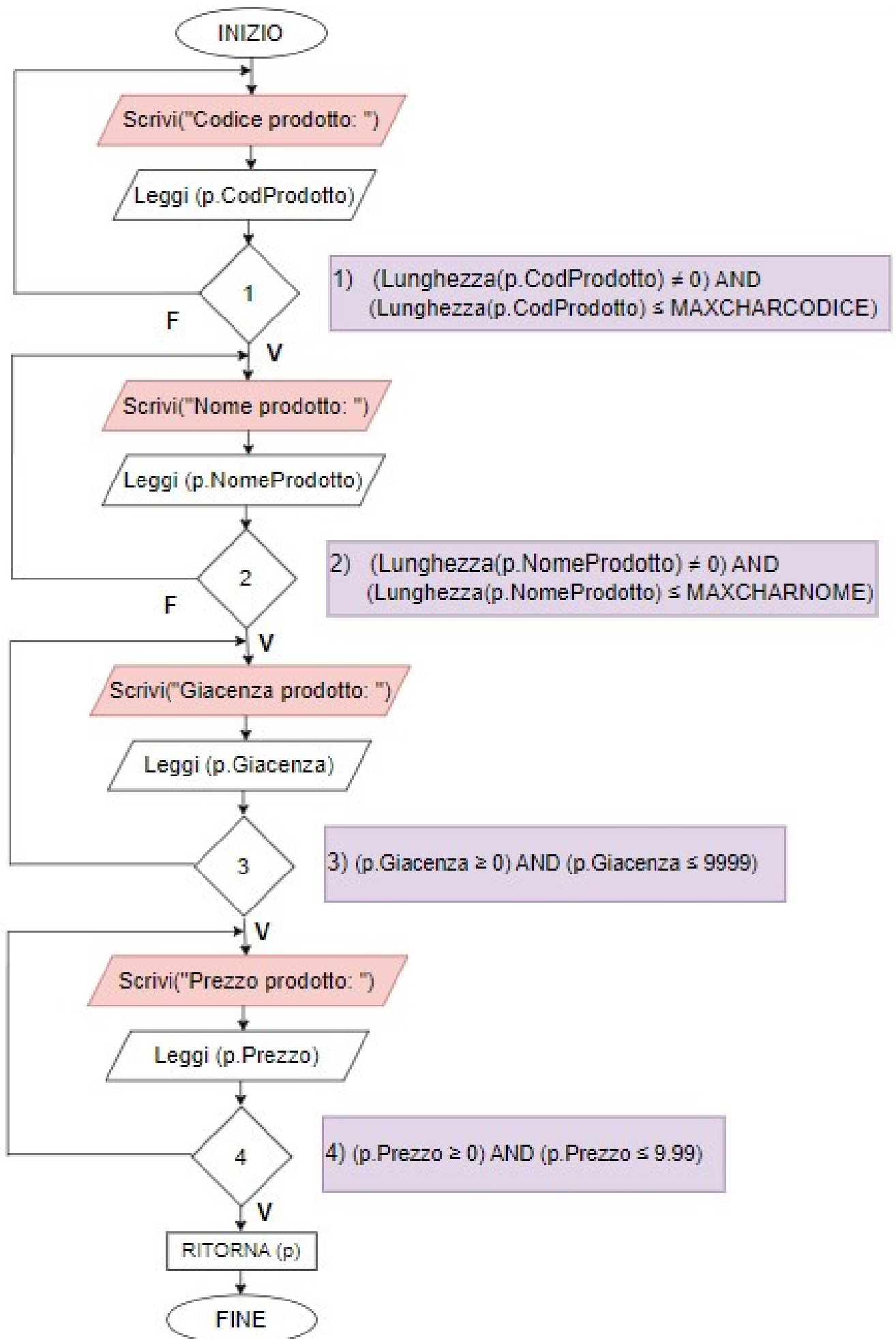
PROCEDURA **main** ()



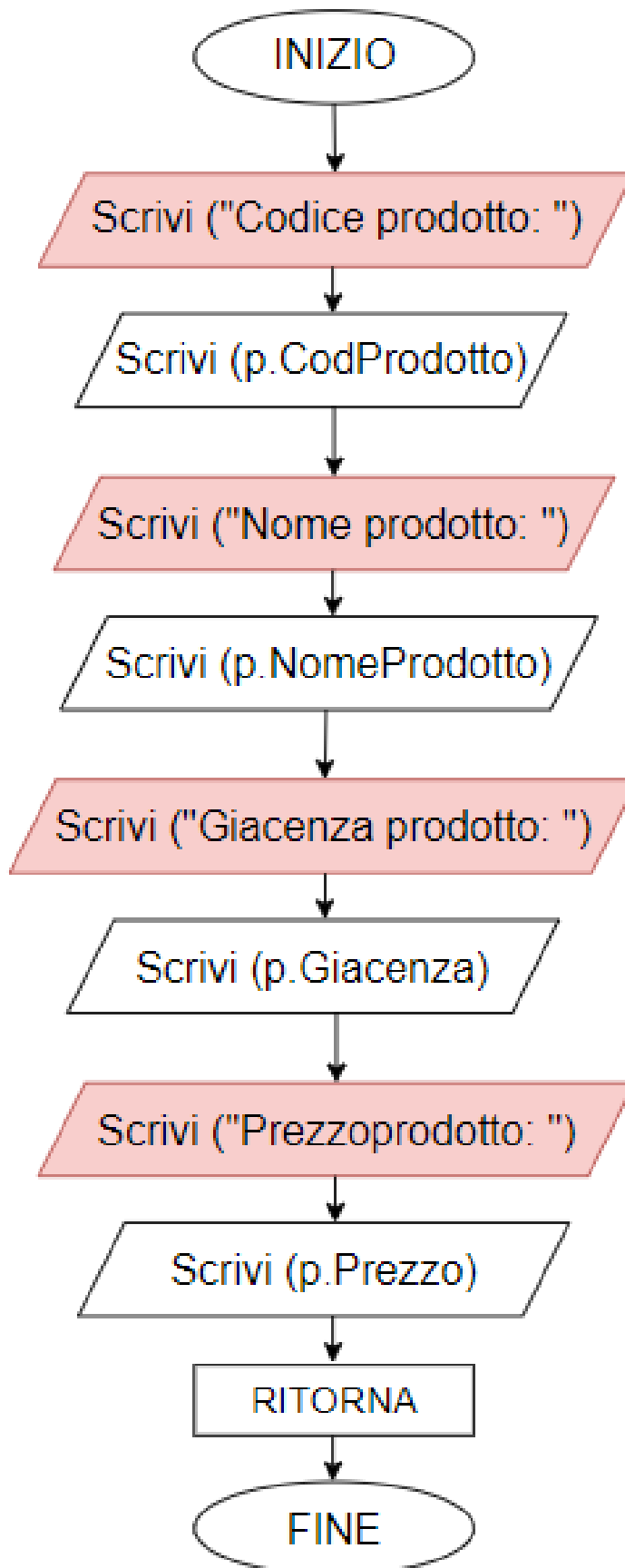
PROCEDURA `Carica_Record_P ()`



ALTERNATIVA: FUNZIONE **Carica_Record_F ()**



PROCEDURA **Visualizza_Record_P** ()



CODIFICA C

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

#define MAXCHARCODICE 6
#define MAXCHARNOME 25

typedef struct
{
    char CodProdotto [MAXCHARCODICE + 1];
    char NomeProdotto [MAXCHARNOME + 1];
    int Giacenza;
    float Prezzo;
} Prodotto;

// prototipi funzioni C necessarie
void Carica_Record_P (Prodotto *p);
Prodotto Carica_Record_F ();
void Visualizza_Record_P (Prodotto p);

// PROCEDURA main()
int main(int argc, char*argv[])
{
    /* dati di input/output */
    Prodotto p;

    /* CALL alla PROCEDURA Carica_Record_P() */
    Carica_Record_P (&p);

    /* CALL alla FUNZIONE Carica_Record_F() */
    //p = Carica_Record_F();

    /* CALL alla PROCEDURA Visualizza_Record_P() */
    Visualizza_Record_P (p);

    return 0;
}

// PROCEDURA Carica_Record_P()
void Carica_Record_P (Prodotto *p)
{
    // caricamento e controllo campi del record
    do
    {
        printf ("Codice prodotto: ");
        gets ((*p).CodProdotto);
    }
    while ( (strlen((*p).CodProdotto) == 0) || (strlen((*p).CodProdotto) > MAXCHARCODICE)
    );

    do
    {
        printf ("Nome prodotto: ");
        gets ((*p).NomeProdotto);
    }
    while ( (strlen((*p).NomeProdotto) == 0) || (strlen((*p).NomeProdotto) > MAXCHARNOME)
    );
}
```

```

do
{
printf ("Giacenza prodotto: ");
scanf("%d", &(p->Giacenza)); //alternativa scanf("%d", &((*p).Giacenza));
}
while ( ((*p).Giacenza < 0) || ((*p).Giacenza > 9999) );

do
{
printf ("Prezzo prodotto: ");
scanf("%f", &(p->Prezzo)); //alternativa scanf("%f", &((*p).Giacenza));
}
while ( ((*p).Prezzo < 0) || ((*p).Prezzo > 9.99) );

return;
}

// FUNZIONE Carica_Record_F()
Prodotto Carica_Record_F ( )
{
Prodotto p;

// caricamento e controllo campi del record
do
{
printf ("Codice prodotto: ");
gets (p.CodProdotto);
}
while ( (strlen(p.CodProdotto) == 0) || (strlen(p.CodProdotto) > MAXCHARCODICE) );

do
{
printf ("Nome prodotto: ");
gets (p.NomeProdotto);
}
while ( (strlen(p.NomeProdotto) == 0) || (strlen(p.NomeProdotto) > MAXCHARNOME) );

do
{
printf ("Giacenza prodotto: ");
scanf("%d", &(p.Giacenza)); //alternativa scanf("%d", &(p.Giacenza));
}
while ( (p.Giacenza < 0) || (p.Giacenza > 9999) );

do
{
printf ("Prezzo prodotto: ");
scanf("%f", &(p.Prezzo)); //alternativa scanf("%f", &(p.Prezzo));
}
while ( (p.Prezzo < 0) || (p.Prezzo > 9.99) );

return (p);
}

```

```
// PROCEDURA Visualizza_Record_P()
void Visualizza_Record_P (Prodotto p)
{

printf ("\n***** STAMPA *****");
// Visualizzazione campi del record
// Codice prodotto
printf ("\nCodice prodotto: ");
puts (p.CodProdotto);

// Nome prodotto
printf ("Nome prodotto: ");
puts (p.NomeProdotto);

// Giacenza prodotto
printf ("Giacenza prodotto: ");
printf ("%d",p.Giacenza);

// Prezzo prodotto
printf ("\nPrezzo prodotto: ");
printf (".2f",p.Prezzo);

return;
}
```