STRUCT - ESERCIZIO 1

Si vuole realizzare un programma che lavora su un insieme di prodotti. Ogni prodotto e' descritto da una struttura contenente il codice (intero) e il prezzo (float).

Il programma deve

- Leggere un array di massimo 6 strutture prodotto. Ogni volta che si effettua una nuova lettura si controlli che il vettore NON contenga gia' dati sullo stesso prodotto, ossia che non ci sia gia' una cella del vettore contenente il medesimo codice prodotto.
- Legga da utente un codice Cod e un prezzo P e li passi a una procedura

```
void mediaprezzo (prodotto V[], int dim, int Cod, float P);
```

che modifichi la cella del vettore V corrispondente a Cod in modo da mettere nel campo prezzo la media tra il vecchio e il nuovo prezzo P.

STRUCT - ESERCIZIO 1

```
#include <stdio.h>
#define N 3

typedef struct{int codice;
float prezzo;} prodotto;

void mediaprezzo(prodotto Prod[], int dim, int Cod, float P)
{int i=0, trovato = 0;
  while ((i<dim) && (!trovato))
    {if (Prod[i].codice == Cod)
      {Prod[i].prezzo = (Prod[i].prezzo + P)/2;
      trovato = 1;}

      i++;
    }
}
```

STRUCT - ESERCIZIO 1

```
void main()
{ prodotto V[N];
  int i=0,j=0, cod, trovato = 0;
  float prezzo;
  while (i<N)      /* LETTURA */
    {printf("inserire codice\n");
      scanf("%d", &cod);
      printf("inserire prezzo\n");
      scanf("%f", &prezzo);
      /* CONTROLLO SE IL DATO E' PRESENTE */
      j=0; trovato = 0;
      while ((j<i)&& (!trovato))
        {if(V[j].codice == cod)
          trovato = 1;
          j++;}
      if (!trovato) /* DATO NON PRESENTE */
        {V[i].codice = cod;
          V[i].prezzo = prezzo;
          i++;}
      else /* DATO GIA' PRESENTE */
        printf("l'elemento e' gia' presente\n");
    }
}
```

STRUCT - ESERCIZIO 1

```
/* E' BUONA NORMA STAMPARE IL VETTORE LETTO PER CORREGGERE
EVENTUALI ERRORI */
printf("Vettore\n");
for (i=0; i<N; i++)
    {printf("%d\n",V[i].codice);
      printf("%f\n",V[i].prezzo);
    }

printf("Inserisci un nuovo codice \n");
scanf("%d", &cod);
printf("Inserisci un nuovo prezzo \n");
scanf("%f", &prezzo);
mediaprezzo(V,N,cod,prezzo);

/* STAMPA VETTORE MODIFICATO */
printf("Nuovo vettore\n");
for (i=0; i<N; i++)
    {printf("%d\n",V[i].codice);
      printf("%f\n",V[i].prezzo);
    }
}
```

VETTORI- ESERCIZIO 2

Si vuole realizzare un programma che legge da utente 10 numeri interi e li inserisce in maniera ordinata (in senso crescente) in un array.

Ad esempio se l'utente inserisce:

4 1 2 5 3 6 8 7 9 0

il vettore stampato e'

0 1 2 3 4 5 6 7 8 9

VETTORI- ESERCIZIO 2

```
#include <stdio.h>
#define N 10

int trovaposizione(int V[],int dim,int num);
void spostasuccessivi(int V[], int pos, int inseriti);

void main()
{int V[N];
  int i, inseriti=0,num,pos;
  for(i=0;i<N;i++)
    { printf("inserisci numero");
      scanf("%d", &num);
      pos=trovaposizione(V,inseriti,num);
      printf("posizione :  %d",pos);
      spostasuccessivi(V,pos,inseriti);
      inseriti ++;
      V[pos]=num;
    }
  printf("vettore\n");
  for(i=0;i<N;i++)
    printf("%d\n",V[i]);
}
```

VETTORI- ESERCIZIO 2

```
int trovaposizione(int V[],int dim,int num)
{int i=0, trovato =0,pos;
  while ((i<dim) && (!trovato))
    {if (num < V[i])
      {pos = i;
       trovato = 1;
      }
     i++;
    }
  if (! trovato) pos=dim;
return pos;

}
```

```
void spostasuccessivi(int V[], int pos, int inseriti)
{int i=0;
  for (i = inseriti-1; i>=pos; i--)
    V[i+1] = V[i];
}
```