

Tipi di dati definiti dall'utente

Tipi enumerativi

Dott.ssa Veronica Rossano
rossano@di.uniba.it
<http://www.di.uniba.it/~rossano>

Tipi definiti dall'utente

- In C è possibile definire nuovi tipi di dati che consentono di rappresentare il mondo reale del problema in maniera più accurata

Tipi enumerativi

- Un tipo enumerativo consente di definire un tipo di dato che può assumere valori solo in un insieme finito di valori
- La sintassi per la definizione del tipo di dato è la seguente:

```
typedef enum  
{ valore1, valore2, ..., valoreN }  
nome_tipo;
```

- La sintassi per la definizione della variabile è la seguente:

```
nome_tipo nome_variabile;
```

Esempio

```
typedef enum  
{lunedì, martedì, mercoledì, giovedì, venerdì, sabato, domenica}  
settimana;  
  
settimana giorno; /*dichiarazione di una  
variabile*/
```

- La definizione di un tipo di dato enumerativo consente di utilizzare gli operatori relazionali e aritmetici sulle variabili dichiarate di tale tipo

```
lunedì<domenica  
giorno!=mercoledì  
lunedì<=giorno && giorno<=venerdì
```

Tipi di dati definiti dall'utente

Array

Dott.ssa Veronica Rossano
rossano@di.uniba.it
<http://www.di.uniba.it/~rossano>

Dichiarazione di un vettore

- La dichiarazione di un array richiede la specifica del tipo di dato e del numero di dati che la struttura deve contenere.
- La sintassi per un array monodimensionale è la seguente:

```
tipo_dato nome_vettore[num_elementi];
```

- `double x[8];`
- `char nome[20];`
- `int mesi[12];`

Array

- Un array è una collezione di due o più celle di memoria adiacenti chiamate **elementi**
- Gli array possono essere:
 - Monodimensionali, detti anche **vettori**
 - Multidimensionali, detti anche **tabelle** o **matrici**

Usare un vettore

- Per riferirsi a ciascun elemento memorizzato in un array è necessario semplicemente indicare la posizione dell'elemento nel vettore secondo la sintassi

```
nome_vettore[indice];
```

- `x[3];`
/*si riferisce al quarto elemento del vettore x*/
- `nome[10];`
/*si riferisce all'11 lettera del vettore nome*/
- `mesi[11];`
/*si riferisce dicembre */

Il valore
dell'indice parte
da **0**
Il primo
elemento del
vettore è
vett[0]

Usare gli indici...

x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]
12.2	23	52.3	1	89.23	4.25	10.1	-563

1. `float x[8] ; i = 5 ;`
2. `x[i - 1] = x[i] ;`
3. `x[i] = x[i + 1] ;`
4. `printf ("%d %.1f", i, x[i]) ;`
5. `printf ("%%.1f", x[--i]) ;`
6. `printf ("%%.1f", x[i++]) ;`
7. `printf ("%%.1f", x[2 * i - 3]) ;`

Output

```
x[4]=x[5]=4.25
x[5] = x[6]=10.1
Printf 5 e 10.1
Printf x[4]= 4.3
Printf x[4]=4.3
Printf x[7]=-563
```

Attenzione solo le istruzioni n. 5 e 6 provocano una modifica dell'indice i

Acquisizione degli elementi

- Avvalorare gli elementi di un vettore equivale ad assegnare un valore ad una variabile qualsiasi in un programma
- La memorizzazione di un valore può avvenire tramite operazioni di:
 - assegnazione
 - input

Esempio

- Scrivere un algoritmo e il relativo programma che acquisisca gli elementi di un vettore come input digitati da tastiera dall'utente

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10

int main ()
{
    int vettore[MAX];
    int i;
    for (i=0; i<MAX; i++)
    {
        printf("\nInserire il %d valore --> ", (i+1));
        scanf("%d", &vettore[i]);
    }
    system("pause");
    return(0);
}
```

Riempivettore.c

Costante per la dimensione del vettore

Dichiarazione del vettore

Iterazione limitata per avvalorare gli elementi del vettore

Inizializzare un vettore

- È possibile dichiarare un vettore mediante inizializzazione seguendo la seguente sintassi

```
tipo_dato nome_vettore[] = {dato1, dato2,...,datoN};
```

- `char vocali[] = {'a', 'e', 'i', 'o', 'u'};`
- `int mesi[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};`

Scandire un vettore

- Per visualizzare i singoli elementi di un vettore è necessario utilizzare una iterazione limitata che scandisca il vettore dal primo all'ultimo elemento

VisualizzaVettore.c

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10
```

```
int main ()
{
    int i;
    int vettore[] = {0, 1, 1, 2, 3, 5, 8, 13, 21, 34};
    for (i=0; i<MAX; i++)
    {
        printf("Il %d valore e' --> %d\n", (i+1), vettore[i]);
    }
    system("pause");
    return(0);
}
```

Inizializzazione del vettore

Iterazione limitata per visualizzare gli elementi del vettore

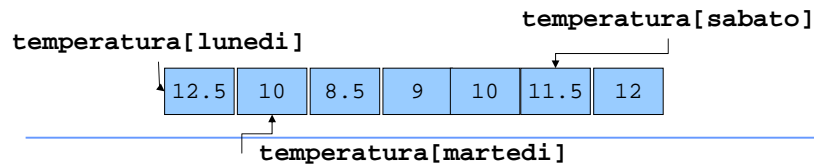
Vettori e tipi enumerativi

- È possibile definire un vettore che abbia come indice un tipo enumerativo
- È necessario assicurarsi che il numero degli elementi del vettore sia uguale alla cardinalità degli elementi del tipo enumerativo

Esempio

- Definire un vettore che raccolga le temperature per ciascun giorno della settimana

```
typedef enum
{
    lunedì, martedì, mercoledì, giovedì, venerdì,
    sabato, domenica
}
settimana;
float temperatura[7];
```



Scandire un vettore enumerativo

- Anche per leggere gli elementi di un vettore enumerativo si usa una iterazione limitata come indicato di seguito:

```
typedef enum
{
    lunedì, martedì, mercoledì, giovedì,
    venerdì, sabato, domenica
}
settimana;
float temperatura[7];
settimana i;
for(i=lunedì; i<=domenica; i++)
{
    scanf("%f", &temperatura[i]);
}
```

Esempio

- Scrivere un algoritmo e il relativo programma che memorizzi in un vettore le temperature di una settimana e ne restituisca la media

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

typedef enum
{
    lunedì, martedì, mercoledì, giovedì, venerdì, sabato, domenica
}
settimana;

int main ()
{
    float temperatura[7];
    float media=0;
    settimana i;

    printf("\n\n***** Il programma calcola la media della temperatura di una settimana *****\n\n");
    for(i=lunedì; i<=domenica; ++i)
    {
        printf("Inserisci la temperatura di ");
        switch (i)
        {
            case lunedì: printf("lunedì -->");
                        break;
            case martedì: printf("martedì -->");
                        break;
            case mercoledì: printf("mercoledì -->");
                        break;
            case giovedì: printf("giovedì -->");
                        break;
            case venerdì: printf("venerdì -->");
                        break;
            case sabato: printf("sabato -->");
                        break;
            case domenica: printf("domenica -->");
                        break;
        }
        scanf("%f", &temperatura[i]);
        media+=temperatura[i];
    }
    media/=7;
    printf("La temperatura media della settimana e' stata : %.2f\n", media);
    system("pause");
}
```

Dichiarazione Tipo di dato
enumerativo

Dichiarazione indice del
vettore

Scansione del vettore

Esercizio

- Scrivere un algoritmo e il relativo programma che generi e memorizzi i primi $n \geq 1$ (definito in input dall'utente) elementi della successione di Fibonacci in un vettore e che lo visualizzi all'utente
 - I primi due termini sono 0 e 1
 - Ogni termine successivo è ottenuto come
 - somma degli ultimi 2 termini
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...