

Fondamenti di Informatica e Laboratorio T-AB e Fondamenti di Informatica T1
Ingegneria Elettronica e Telecomunicazioni e
Ingegneria dell'Automazione
a.a. 2010/2011

Lab 06

Array

Esercizio 1

- Creare un programma che legga da input un numero non noto a priori di interi (al più 10) terminati da 0. Tale sequenza può eventualmente contenere numeri ripetuti.
- Si memorizzi tale sequenza in un vettore di opportuna dimensione.
- Si stampino a video tutti i numeri per cui il successivo nel vettore è pari al numero stesso

Esercizio 1 – Soluzione

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

int main(void) {
    int num, size, i;
    int values[DIM];

    size = 0;
    do {
        printf("Inserisci un numero:");
        scanf("%d", &num);
        if (num!=0 && size<DIM) {
            values[size] = num;
            size++;
        }
    } while (num!=0 && size<DIM);

    for (i=0; i<size-1; i++)
        if (values[i] == values[i+1])
            printf("%d ", values[i]);
    return (0); }
```

Esercizio 2

- Creare un programma che legga da input un numero non noto a priori di interi (al più 10) terminati da 0.
- Si memorizzi tale sequenza in un vettore di opportuna dimensione, rispettando l'ordine con cui i valori sono stati inseriti.
- Si memorizzino poi in un secondo vettore i valori del primo, avendo cura di inserirli in ordine inverso
- Si stampi infine il vettore coi valori in ordine invertito

Esercizio 2 – Soluzione

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

int main(void) {
    int num, size, i, j;
    int values[DIM], inv[DIM];
    size = 0;
    do {
        printf("Inserisci un numero:");
        scanf("%d", &num);
        if (num>0 && size<DIM) {
            values[size] = num;
            size++;
        }
    } while (num!=0 && size<DIM);

    j=0;
    for (i=size-1; i>=0; i--) {
        inv[j] = values[i];
        j++;
    }
    for (i=0; i<size; i++) printf("%d ", inv[i]);
    system("PAUSE");
    return (0);
}
```

Esercizio 3

- Creare un programma che legga da input una sequenza di interi positivi. In particolare, l'utente inserisce un numero iniziale con cui specifica quanti numeri è intenzionato ad inserire (al max 10). Di seguito poi inserisce i numeri, tutti in ordine strettamente crescente.
- In fase di lettura, il programma controlli che ogni numero sia effettivamente maggiore del precedente (si scartino i valori che non rispettano tale criterio).
- In un secondo vettore si calcoli la differenza percentuale tra un valore ed il successivo (data dalla differenza tra i due valori, divisa poi per il primo valore e moltiplicato il tutto per 100)
- Si richieda poi all'utente un valore di soglia (in percentuale) , e si stampino a video tutte le coppie di valori il cui aumento dal primo al secondo valore risulta essere, in percentuale, maggiore della soglia specificata

Esercizio 3 – Soluzione

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

int main(void) {
    int num, size, i, prec;
    int values[DIM];
    float soglia;
    float ratio[DIM];

    do {
        printf("Quanti numeri vuoi inserire ? (MAX 10) ");
        scanf("%d", &size);
    } while (size < 0 || size > DIM);

    prec = -1;
    for (i=0; i<size; ) {
        printf("Inserisci un numero:");
        scanf("%d", &num);
        if ( num > prec ) {
            values[i] = num;
            prec = num;
            i++;
        }
    }
    ...
}
```

Esercizio 3 – Soluzione

...

```
for (i=0; i<size-1; i++)
    ratio[i] = ( (values[i+1]-values[i]) / ((float) values[i])) * 100;

printf("Inserire soglia: ");
scanf("%f", &soglia);
for (i=0; i<size-1; i++)
    if (ratio[i] > soglia)
        printf("%d %d\n", values[i], values[i+1]);

system("PAUSE");
return (0);
}
```


Esercizio 4

Scrivere un programma che, dato un vettore **NUM** di **N** interi positivi inseriti dall'utente, ne produca due **PAR** e **DIS** contenenti, rispettivamente, i numeri pari e dispari del vettore iniziale.

Si controlli che i numeri inseriti dall'utente siano positivi.

Per verificare se un numero è pari, si scriva una funzione che restituisca 1 se il numero è pari, 0 altrimenti

```
int pari(int n) ;
```

Esercizio 4

Esempio: l'utente inserisce la sequenza

4 2 **-4** 5 6 **-9** 1 6

└─┐ *scartati* └─┐

NUM

4	2	5	6	1	6
---	---	---	---	---	---

PAR

4	2	6	6		
---	---	---	---	--	--

DIMENSIONE LOGICA 4

DIS

5	1				
---	---	--	--	--	--

DIMENSIONE LOGICA 2

Esercizio 4

```
#include <stdio.h>
#define N 8

int pari(int n){
    if (n%2 == 0) return 1;
    else return 0;
}

void main()
{int num[N], par[N], dis[N];
  int i,ivp=0,ivd=0; /* ivp = indice vett pari e ivd = indice vett
dispari*/

  for (i=0; i<N; i++) /*lettura vettore num */
    do {printf("inserire intero positivo del vettore NUM ");
        scanf("%d", &num[i]);
        }
    while (num[i] < 0);
```

Continua...

Esercizio 4

```
for (i=0; i<N; i++) /* trasferimento pari in PAR e dispari in DIS*/
    if (pari(num[i])){
        par[ivp] = num[i];
        ivp++;}
    else { dis[ivd] = num[i];
        ivd++;
    }

/* stampa dei vettori PAR e DIS
    ATTENZIONE !!!!! La dimensione logica di PAR e' ivp mentre
                        la dimensione logica di DIS e' ivd */

for (i=0; i<ivp; i++)
    printf("%d  ", par[i]);

printf("\n");

for (i=0; i<ivd; i++)
    printf("%d  ", dis[i]);

}
```

Esercizio 5

- Creare un programma che legga da input un numero non noto a priori di interi positivi (al più 10) terminati da 0.
- Qualora l'utente inserisca dei valori negativi, tali valori devono essere scartati e non considerati
- Si memorizzi tale sequenza in un vettore di opportuna dimensione.
- Si stampino a video tutti i numeri che sono allocati nel vettore in posizioni il cui indice è uguale al numero stesso

Esercizio 5 – Soluzione

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

int main(void) {
    int num, size, i;
    int values[DIM];

    size = 0;
    do {
        printf("Inserisci un numero:");
        scanf("%d", &num);
        if (num>0 && size<DIM) {
            values[size] = num;
            size++;
        }
    } while (num!=0 && size<DIM);

    for (i=0; i<size; i++)
        if (values[i] == i)
            printf("%d ", values[i]);
    system("PAUSE");
    return (0); }
```

Esercizio 6

- Si vogliono elaborare alcuni dati metereologici, riguardanti alcune località sciistiche. Per ogni località, un'utente inserisce il codice della località e i cm. di manto nevoso (entrambi interi). Il codice 0, inserito come località, indica che l'utente non vuole inserire altri dati.
- Il programma deve memorizzare tali dati in due appositi vettori (uno per le località ed uno per la neve caduta)
- Il programma deve poi stampare a video i codici di tutte le località che risultino avere un manto nevoso inferiore alla media, calcolata sui valori inseriti

Esercizio 6 – Soluzione

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

int main(void) {
    int cod_loc, size, i, tot;
    int loc[DIM], neve[DIM];
    float media;

    size = 0;
    do {
        printf("Inserisci codice localita': ");
        scanf("%d", &cod_loc);
        if ( cod_loc != 0 && size < DIM ) {
            loc[size] = cod_loc;
            printf("Manto nevoso (cm.): ");
            scanf("%d", &(neve[size]));
            size++;
        }
    } while (cod_loc!=0 && size<DIM);
```

...

Esercizio 6 – Soluzione

...

```
tot = 0;
for (i=0; i<size; i++)
    tot = tot + neve[i];
media = tot / ((float) size);

for (i=0; i< size; i++)
    if (neve[i]<media)
        printf("localita' con poca neve: %d\n", loc[i]);

system("PAUSE");
return (0);
}
```