

# Fondamenti di Informatica

---

## Lab 06 Array2

# Esercizio 1

- Creare un programma che legga da input una sequenza di interi, di lunghezza non nota a priori (al più 10), e terminata da 0. A tal scopo, si realizzi una funzione che riceva come parametri di ingresso un vettore e la sua dimensione fisica, e restituisca la dimensione logica del vettore. Tale funzione si deve fare carico della fase di lettura e riempimento dell'array. La sequenza può contenere elementi ripetuti (anche più volte).

```
int leggi(int vet[], int dim) {...}
```

- Si realizzi una funzione che, ricevuti un array e la sua dimensione, ed un elemento da cercare, restituisca "vero" se l'elemento è presente nell'array.

```
int trova(int vet[], int dim, int el) {...}
```

- Si realizzi una funzione che, ricevuti in ingresso il primo vettore con la sua dimensione logica, ed un secondo vettore con la sua dimensione fisica, memorizzi nel secondo vettore tutti gli elementi del primo, ma senza ripetizioni. La funzione restituisca la dimensione logica del secondo vettore. A tal scopo si utilizzi la funzione di cui al punto precedente.

```
int eliminaRipetuti(int vet[], int dim_v, int single[],  
int dim_s) {...}
```

- Si realizzi un main che invoca le funzioni, e che stampi a video l'elenco degli elementi non ripetuti

# Esercizio 1 – Soluzione

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

int leggi(int vet[], int dim) {
    int i, num;

    i=0;
    do {
        printf("Inserisci numero: ");
        scanf("%d", &num);
        if (num != 0) {
            vet[i] = num;
            i++;
        }
    } while (num!=0 && i<dim);
    return i;
}
```

...

# Esercizio 1 – Soluzione

```
int trova(int vet[], int dim, int el) {
    int trovato, i;

    trovato = 0;
    for (i=0; i<dim && !trovato; i++) {
        if (vet[i] == el)
            trovato = 1;
    }
    return trovato;
}

int eliminaRipetuti(int vet[], int dim_v, int single[], int dim_s) {
    int i, j;

    j = 0;
    for (i=0; i<dim_v && j<dim_s; i++) {
        if (!trova(single, j, vet[i])) {
            single[j] = vet[i];
            j++;
        }
    }
    return j;
}
...
```

# Esercizio 1 – Soluzione

```
int main(void)
{
    int v1[10], v2[10], i;
    int dim_v1, dim_v2;

    dim_v1 = leggi(v1, 10);
    dim_v2 = eliminaRipetuti(v1, dim_v1, v2, 10);

    for (i=0; i<dim_v2; i++)
        printf("%d ", v2[i]);

    system("PAUSE");

    return (0);
}
```

# Esercizio 2

- Creare un programma che legga da input una sequenza di interi, di lunghezza non nota a priori (al più 10), e terminata da 0. A tal scopo, si realizzi una funzione che riceva come parametri di ingresso un vettore e la sua dimensione fisica, e restituisca la dimensione logica del vettore. Tale funzione si deve fare carico della fase di lettura e riempimento dell'array. La sequenza può contenere elementi ripetuti (anche più volte).

```
int inserimento(int array[], int dim_f);
```

- Si realizzi una funzione che, ricevuti un array e la sua dimensione, ed un elemento da cercare, *restituisca il valore -1 se l'elemento non è presente nell'array; altrimenti restituisca il primo indice in cui è presente l'elemento cercato.*

```
int ricerca(int array[], int dim_l, int elem);
```

- Si realizzi un main che invoca le funzioni, e che stampi a video l'elenco degli elementi che compaiono più volte nel vettore

# Esercizio 2 – Soluzione

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

int leggi(int vet[], int dim) {
    int i, num;

    i=0;
    do {
        printf("Inserisci numero: ");
        scanf("%d", &num);
        if (num != 0) {
            vet[i] = num;
            i++;
        }
    } while (num!=0 && i<dim);
    return i;
}
```

...

# Esercizio 2 – Soluzione

...

```
int trovaPos(int vet[], int dim, int el) {  
    int trovato, i;  
  
    trovato = -1;  
    for (i=0; i<dim && trovato<0; i++) {  
        if (vet[i] == el)  
            trovato = i;  
    }  
    return trovato;  
}
```

...

# Esercizio 2 – Soluzione

```
int main(void)
{

    int v[10], dim, i;

    dim = leggi(v, 10);
    for (i=0; i<dim; i++)
        if ( trovaPos(&v[i+1], dim-i-1, v[i]) >= 0 )
            if ( trovaPos(v, i, v[i]) < 0 )
                printf("%d ", v[i]);

    system("PAUSE");

    return (0);
}
```

# Esercizio 3

- Creare un programma che legga da input due sequenze di interi, di lunghezza non nota a priori (al più 10), e terminate da 0. A tal scopo, si realizzi una apposita funzione.

```
int inserimento(int array[], int dim_f)
```

- Si realizzi una funzione che, ricevuti un array e la sua dimensione, ed un elemento da cercare, *restituisca il valore -1 se l'elemento non è presente nell'array; altrimenti restituisca il primo indice in cui è presente l'elemento cercato.*

```
int ricerca(int array[], int dim_1, int elem);
```

- Si realizzi un main che invoca le funzioni, e che stampi a video tutti gli elementi del primo vettore che NON compaiono nel secondo.

# Esercizio 3 – Soluzione

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

int leggi(int vet[], int dim) {
    int i, num;

    i=0;
    do {
        printf("Inserisci numero: ");
        scanf("%d", &num);
        if (num != 0) {
            vet[i] = num;
            i++;
        }
    } while (num!=0 && i<dim);
    return i;
}
```

...

# Esercizio 3 – Soluzione

...

```
int trovaPos(int vet[], int dim, int el) {
    int trovato, i;

    trovato = -1;
    for (i=0; i<dim && trovato<0; i++) {
        if (vet[i] == el)
            trovato = i;
    }
    return trovato;
}
```

...

# Esercizio 3 – Soluzione

...

```
int main(void)
{
    int v1[10], v2[10], i;
    int dim_v1, dim_v2;

    dim_v1 = leggi(v1, 10);
    dim_v2 = leggi(v2, 10);

    for (i=0; i<dim_v1; i++)
        if (trovaPos(v2, dim_v2, v1[i]) < 0)
            printf("%d ", v1[i]);

    system("PAUSE");

    return (0);
}
```

# Esercizio 4

- Creare un programma che legga da input due sequenze di interi, di lunghezza non nota a priori (al più 10), e terminate da 0. A tal scopo, si realizzi una apposita funzione.
- Si ipotizzi che le sequenze di numeri inseriti siano ordinate in maniera crescente.
- Si realizzi un main che invochi la funzione per leggere due sequenze, e che stampi a video, in ordine crescente, tutti gli elementi di entrambi i vettori. Ad esempio, con  $v1=\{1,3,5,7\}$  e  $v2=\{2,4,6,8\}$  il programma deve stampare: 1,2,3,4,5,6,7,8

# Esercizio 4 – Soluzione

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

int leggi(int vet[], int dim) {
    int i, num;

    i=0;
    do {
        printf("Inserisci numero: ");
        scanf("%d", &num);
        if (num != 0) {
            vet[i] = num;
            i++;
        }
    } while (num!=0 && i<dim);
    return i;
}
```

...

# Esercizio 4 – Soluzione

```
int main(void) {
    int v1[10], v2[10], i, j;
    int dim_v1, dim_v2;

    dim_v1 = leggi(v1, 10);
    dim_v2 = leggi(v2, 10);

    i = 0;
    j = 0;
    while (i<dim_v1 && j<dim_v2) {
        if (v1[i] < v2[j]) {
            printf("%d ", v1[i]);
            i++;
        }
        else {
            printf("%d ", v2[j]);
            j++;
        }
    }
    while (i<dim_v1) {
        printf("%d ", v1[i]);
        i++;
    }
    while (j<dim_v2) {
        printf("%d ", v2[j]);
        j++;
    }
    system("PAUSE"); return (0); }
```

# Esercizio 5

- Creare un programma che legga da input una sequenza di interi, di lunghezza non nota a priori (al più 10), e terminata da 0. A tal scopo, si realizzi una apposita funzione.

```
int inserimento(int array[], int dim_f)
```

- Si realizzi una funzione che, ricevuti un array e la sua dimensione, ed un elemento da cercare, *restituisca il valore -1 se l'elemento non è presente nell'array; altrimenti restituisca il primo indice in cui è presente l'elemento cercato.*

```
int ricerca(int array[], int dim_l, int elem);
```

- Si realizzi un main che invoca le funzioni, e che stampi a video tutti gli elementi del vettore che compaiono esattamente 2 volte nel vettore.

# Esercizio 5 – Soluzione

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

int leggi(int vet[], int dim) {
    int i, num;

    i=0;
    do {
        printf("Inserisci numero: ");
        scanf("%d", &num);
        if (num != 0) {
            vet[i] = num;
            i++;
        }
    } while (num!=0 && i<dim);
    return i;
}
```

...

# Esercizio 5 – Soluzione

...

```
int trovaPos(int vet[], int dim, int el) {
    int trovato, i;

    trovato = -1;
    for (i=0; i<dim && trovato<0; i++) {
        if (vet[i] == el)
            trovato = i;
    }
    return trovato;
}
```

...

# Esercizio 5 – Soluzione

```
int main(void) {
    int v[10], dim, i, count, temp_dim, temp_pos, offset;

    dim = leggi(v, 10);

    for (i=0; i<dim; i++) {
        count = 0;
        temp_dim = dim;
        offset = 0;
        do {
            temp_pos = trovaPos(&v[offset], temp_dim, v[i]);
            if (temp_pos >= 0) {
                count++;
                temp_dim = temp_dim - temp_pos - 1;
                offset = offset + temp_pos + 1;
            }
        } while (temp_pos >= 0);
        if (count == 2)
            printf("L'elemento %d compare 2 volte\n", v[i]);
    }
    system("PAUSE"); return (0); }
```

# Esercizio 6

- Creare un programma che legga da input due sequenze di interi, di lunghezza non nota a priori (al più 10), e terminate da 0. A tal scopo, si realizzi una apposita funzione.
- Si realizzi una funzione che, ricevuti un array e la sua dimensione, ed un elemento da cercare, *restituisca il valore -1 se l'elemento non è presente nell'array; altrimenti restituisca il primo indice in cui è presente l'elemento cercato.*
- Si realizzi un main che invoca le funzioni, e che stampi a video tutti gli elementi del primo vettore che compaiono nel secondo vettore nella stessa posizione (cioè con lo stesso indice). Si supponga per semplicità che le due sequenze abbiano la stessa lunghezza.

# Esercizio 6 – Soluzione

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

int leggi(int vet[], int dim) {
    int i, num;

    i=0;
    do {
        printf("Inserisci numero: ");
        scanf("%d", &num);
        if (num != 0) {
            vet[i] = num;
            i++;
        }
    } while (num!=0 && i<dim);
    return i;
}
```

...

# Esercizio 6 – Soluzione

...

```
int trovaPos(int vet[], int dim, int el) {
    int trovato, i;

    trovato = -1;
    for (i=0; i<dim && trovato<0; i++) {
        if (vet[i] == el)
            trovato = i;
    }
    return trovato;
}
```

...

# Esercizio 6 – Soluzione

```
int main(void) {  
  
    int v1[10], v2[10], i;  
    int dim_v1, dim_v2;  
  
    dim_v1 = leggi(v1, 10);  
    dim_v2 = leggi(v2, 10);  
  
    for (i=0; i<dim_v1; i++)  
        if ( v1[i] == v2[i])  
            printf("%d ", v1[i]);  
  
    system("PAUSE");  
    return (0);  
}
```