

Titolo unità didattica: Array e insiemi

[08]

Titolo modulo : Function in C per operazioni di base su insiemi

[03-C]

Sviluppo di function in C che operano su array che realizzano insiemi ed esempi di utilizzo

Argomenti trattati:

- ✓ function in C per l'unione di due insiemi
- ✓ function in C per l'intersezione di due insiemi
- ✓ function in C per l'inclusione di due insiemi
- ✓ function in C per la sottrazione di due insiemi
- ✓ function in C per l'uguaglianza di due insiemi

Prerequisiti richiesti: AP-07-10-C, AP-08-01-T, AP-08-02-T

esercizio

realizzare una function C per l'unione di due array 1D

```
void unioneF(float a[],int n_a,float b[],int n_b,
            float c[],int *n_c)
{
    int i,i_c;
    /* inizializzazione dell'array c */
    for (i=0;i<n_a;i++)
        c[i] = a[i];
    i_c = n_a-1;
    for (i=0;i<n_b;i++)
    {
        if (!appartiene_aF(b[i],a,n_a)
            {
                i_c++;
                c[i_c]= b[i];
            }
    }
    *n_c = i_c+1;
}
```

```
void unione(in: float a[], int n_a, float b[],
            int n_b; out: float c[], int n_c)
int i;
/* inizializzazione di c */
for (i=0, i < n_a, i++) {
    c(i) = a(i) ;
}
n_c = n_a ;
for (i=1; i < n_b; i++) {
    if ( ! appartiene(b[i],a,n_a) )
    {
        c[n_c] = b[i] ;
        n_c = n_c+1 ;
    }
}
}
```

esercizio

realizzare un main che richiama la function **unioneF** di due array 1D

```
/* programma per l'unione di due insiemi.
   main usa unioneF che usa appartiene_aF */
#include <stdio.h>
int appartiene_aF(float, float [], int); /* function
      sviluppata nel modulo 07-10-C */
void unioneF(float [], int, float [], int, float [], int *);
void main()
{
    int n_a, n_b, n_c;
    float a[]={1.0,20.0,0.5,4.4,9.0,6.1,7.3,3.0,1.1,8.0};
    float b[]={8.0,12.0,7.0,0.5,3.0, 1.1,5.0,1.0};
    float c[18];
    int i;
    n_a = 10;
    n_b = 8;
    unioneF(a, n_a, b, n_b, c, &n_c);
    printf("numero di elementi dell'unione:%d\n", n_c);
    for (i=0; i<n_c; i++)
        printf(" %5.1f\n", c[i]);
}
```

esercizio

realizzare una function C per l'intersezione di due array 1D

```
void intersezioneF(float a[],int n_a,float b[],int n_b,
                  float c[],int *n_c)
{
    int i,i_c;
    i_c = 0;
    for (i=0;i<n_a;i++)
    {
        if (appartiene_aF(a[i],b,n_b))
        {
            c[i_c]= a[i];
            i_c++;
        }
    }
    *n_c = i_c;
}
```

```
void intersezione( in: float a[], int n_a, float b[],
                  int n_b; out: float c[], int n_c) {
    int i ;
    n_c = 0 ;
    for (i=0; i < n_a; i++) {
        if (appartiene(a[i],b,n_b)
        {
            c[n_c] = a[i] ;
            n_c = n_c+1 ;
        }
    }
}
```

esercizio

realizzare un main che richiama la function **intersezioneF** di due array 1D

```
/* programma per l'intersezione di due insiemi.
   main usa intersezioneF che usa appartiene_aF */
#include <stdio.h>
void visualizza_aF (float [],int); /* modulo 07-09-C */
int appartiene_aF(float,float [],int); /* modulo 07-10-C */
void intersezioneF(float [],int,float [],int,float [],int *);
void main()
{
    int n_a,n_b,n_c;
    float a[]={1.0,20.0,0.5,4.4,9.0,6.1,7.3,3.0,1.1,8.0};
    float b[]={8.0,12.0,7.0,0.5,3.0, 1.1,5.0,1.0};
    float c[18];
    int i;
    n_a = 10;
    n_b = 8;
    intersezioneF(a,n_a,b,n_b,c,&n_c);
    printf("numero di elementi dell'intersezione:%d\n",n_c);
    visualizza_aF(c,n_c);
}
```

esercizio

realizzare una function C per l'inclusione di due array 1D

```
int inclusioneF(float a[],int n_a,float b[],int n_b)
{
    int i,incluso;
    i = 0;
    do
        {
            i++;
            incluso = appartiene_aF(a[i],b,n_b);
        }
    while(incluso && i<n_a);
    return incluso;
}
```

```
int inclusioneF(float a[],int n_a,float b[],int n_b)
{
    int i,incluso=1;
    i = 0;
    for(i=0;i<n_a;i++)
        if(!appartiene_aF(a[i],b,n_b))
            {
                incluso = 0;
                break;
            }
    return incluso;
}
```

```

/* programma per l'inclusione di due insiemi.
   main usa inclusioneF che usa appartiene_aF */
#include <stdio.h>
void visualizza_aF (float [],int ); /* modulo 07-09-C */
void legge_da_tastiera_aF(float [],int); /* 07-09-C */
int appartiene_aF(float,float [],int); /* modulo 07-10-C */
int inclusioneF(float [],int,float [],int);
void main()
{
    int i,n_a,n_b;
    float a[100],b[100];
    printf("inserire numero di elementi I array (<=100):");
    scanf("%d",&n_a);
    legge_da_tastiera_aF(a,n_a);
    printf("inserire numero di elementi II array (<=100):");
    scanf("%d",&n_b);
    legge_da_tastiera_aF(b,n_b);
    if(inclusioneF(a,n_a,b,n_b))
        printf("il I array e' incluso nel II array\n",);
    else
        printf("il I array non e' incluso nel II array\n",);
    printf("I array e II array\n",);
    visualizza_aF(a,n_a); visualizza_aF(b,n_b);
}

```

esercizio

realizzare una function C per la sottrazione di due array 1D

```
void sottrazioneF(float a[],int n_a,float b[],int n_b,
                 float c[],int *n_c)
{
    int i,i_c;
    i_c = 0;
    for (i=0;i<n_a;i++)
        if (!appartiene_aF(a[i],b,n_b))
            {
                c[i_c] = a[i];
                i_c++;
            }
    *n_c = i_c;
}
```

```
void sottrazione( in: float a[], int n_a,
                 float b[], int n_b; out: float c[], int n_c) {
    int i ;
    n_c = 0 ;
    for (i=1; i < n_a; i++) {
        if ( ! appartiene(a[i],b,n_b) )
            {
                c[n_c] = a[i] ;
                n_c = n_c+1;
            }
    }
}
```


esercizio

realizzare una function C per determinare l'uguaglianza di due array 1D

```
int uguaglianza_insiemiF(float a[], float b[], int n)
{
    int i, uguale;
    i = 0;
    do
    {
        uguale = appartiene_aF(a[i], b, n);
        i++;
    }
    while (uguale && i < n);
    return uguale;
}
```

```
int uguaglianza_insiemiF(float a[], float
                        b[], int n)
{
    int i;
    i = 0;
    for (i=0; i<n; i++)
        if (!appartiene_aF(a[i], b, n))
            return 0;
    return 1;
}
```

```
logical uguaglianza_insiemi(float a[], float b[],
                           int n) {
    int i;
    logical uguale;
    i = 0;
    do {
        uguale = appartiene(a[i], b, n);
        i = i+1;
    } while (uguale && i < n);
    return uguale;
}
```

Esempio di file di intestazioni per le function su insiemi
`insiemi.h`

```
#include <stdio.h>
void unioneF(float [],int,float [],int,float [],int *);
void intersezioneF(float[],int,float[],int,float[],int*);
int inclusioneF(float [],int,float [],int);
void sottrazioneF(float[],int,float[],int,float[],int *);
int uguaglianza_insiemiF(float [],float [],int);
int appartiene_aF(float,float [],int);
void visualizza_aF (float [],int);
void legge_da_tastiera_aF(float [],int);
```

```
#include "insiemi.h"
void main()
{
    .....
}
```