

Corso di Informatica A.A. 2009-2010

Lezione 16

Esercitazione n. 4

Parte I : Programma sequenza.c

Realizzare un programma che calcoli la media, il minimo e il massimo degli elementi di una sequenza di numeri reali inseriti dall'utente (la fine della sequenza è identificata dal numero 0.0).

Creare a tale scopo due funzioni `media()` e `minmax()` e utilizzare vettori allocati dinamicamente.

```

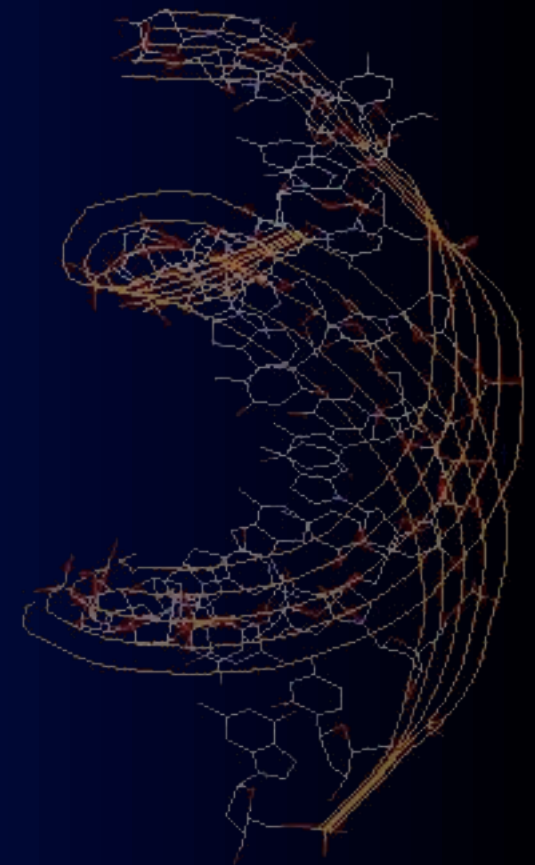
#include <stdio.h>
#include <stdlib.h>

double media(double *vec, int n);
void minmax(double *vec, int n, double *m);

void main( )
{
    double *vettore , elemento , m[2];
    int i , j , fine ;
    vettore = NULL ;
    /* inserimento dati */
    i = 0 ;
    fine = 0 ;
    while (!fine)
    {
        printf( " Inserisci un elemento ( 0 per finire ): " ) ;
        scanf( "%lf",&elemento) ;
        if(elemento == 0.0 )
            fine = 1 ;
        else
        {
            /* aumento di 1 la dimensione del vettore */
            vettore = ( double *) realloc (vettore , (i+1)*sizeof(double) ) ;
            vettore[i] = elemento ;
            i++ ;
        }
    }
    /* calcolo e stampa della media */
    printf ( " La media e' : %lf \n " , media(vettore,i)) ;
    minmax(vettore,i , m);
    printf ( " Il minimo e' %lf \n " , m[0]) ;
    printf ( " Il massimo e' %lf \n " , m[1]) ;

    /* deallocazione del vettore */
    free(vettore) ;
}

```



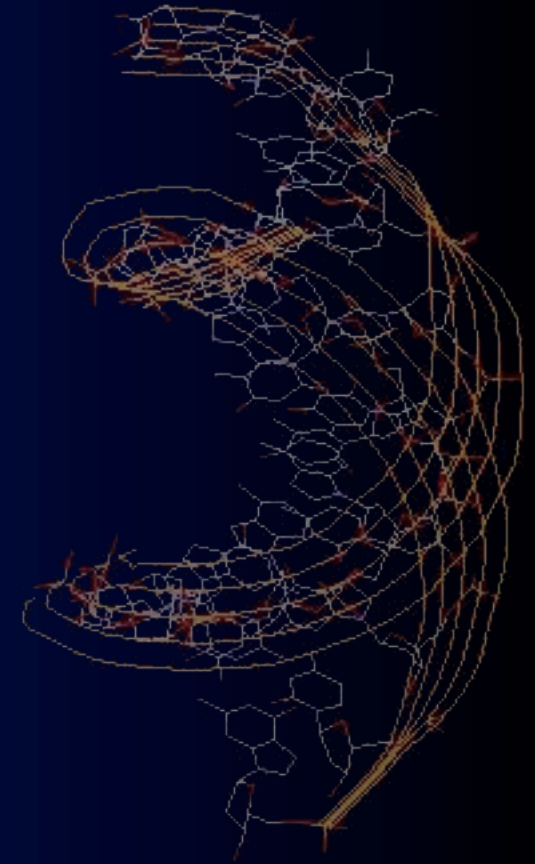
```

double media(double *vec, int n)
{
    int j;
    double somma=0.0,ave;

    for ( j = 0 ; j < n ; j++)
        somma+= vec[j] ;
    ave = somma/n ;
    return ave;
}

void minmax(double *vec, int n, double *m)
{
    int j;
    if(vec[0] >= vec[1])
    {
        m[0]=vec[1];
        m[1]=vec[0];
    }
    else
    {
        m[0]=vec[0];
        m[1]=vec[1];
    }
    for(j=2; j < n; j++)
    {
        if(vec[j] < m[0]) m[0]=vec[j];
        else if(vec[j] > m[1]) m[1]=vec[j];
    }
}

```



Esercitazione n. 4

Parte II : Programma trasposta.c

Realizzare un programma che calcoli la trasposta di una matrice di interi allocata dinamicamente.

Suggerimento

Per costruire una matrice dinamica di r righe e c colonne basterà eseguire le seguenti due operazioni:

1. Allocare dinamicamente un vettore di r puntatori (vettore delle righe).
2. Per ogni riga allocare un vettore di c elementi del tipo base considerato (vettore colonne).

```

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

void main()
{
    int **matrice, **trasposta;
    int righe, colonne, r, c;

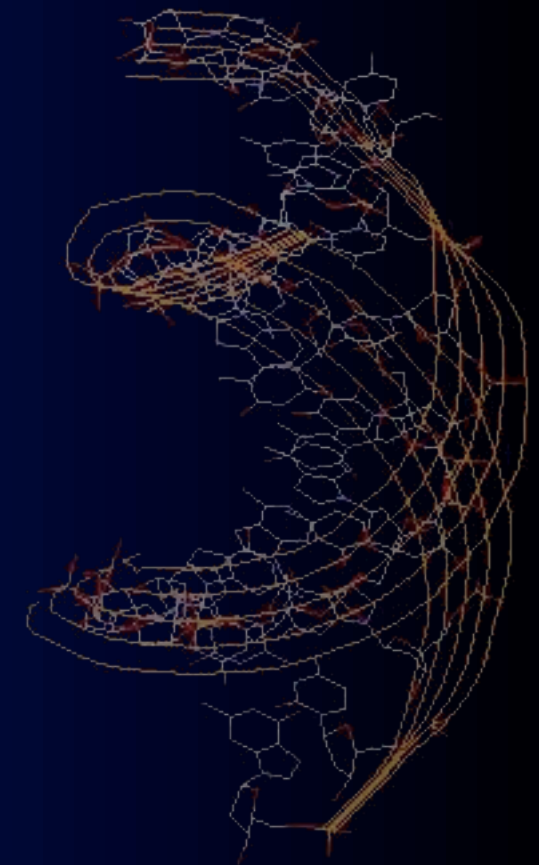
    printf("Quante righe ha la matrice: ");
    scanf("%d", &righe);
    printf("Quante colonne ha la matrice: ");
    scanf("%d", &colonne);

    /* allocazione della matrice */
    matrice = (int **)malloc(righe * sizeof(int *));
    for (r=0; r<righe; r++)
        matrice[r] = (int *)malloc(colonne * sizeof(int));

    /* inserimento dati */
    for (r=0; r<righe; r++)
        for (c=0; c<colonne; c++)
        {
            printf("Inserisci elemento di riga %d e colonna %d: ", r, c);
            scanf("%d", &matrice[r][c]);
        }

    /* alloco la matrice trasposta */
    trasposta = (int **)malloc(colonne * sizeof(int *));
    for (c=0; c<colonne; c++)
        trasposta[c] = (int *)malloc(righe * sizeof(int));

```



```

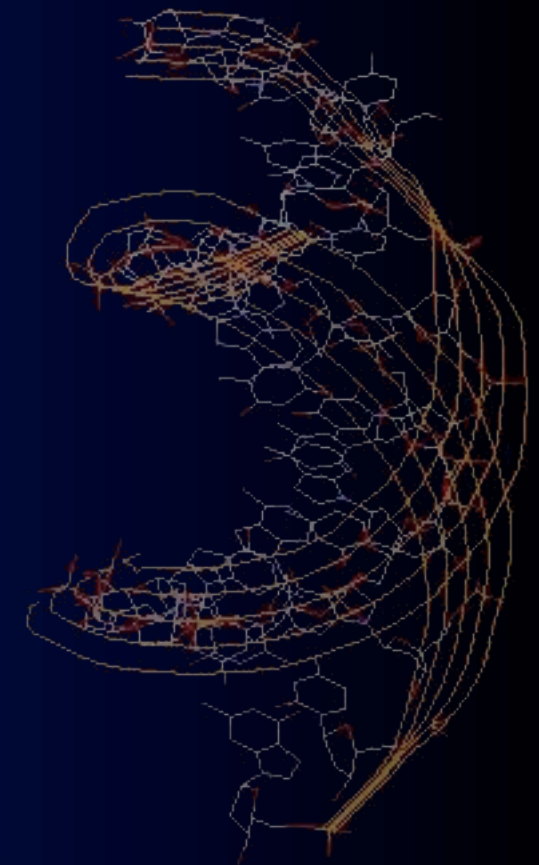
/* calcolo della trasposta */
for (r=0; r<righe; r++)
    for (c=0; c<colonne; c++)
        trasposta[c][r] = matrice[r][c];

/* stampo la trasposta */
printf("La trasposta e':\n");
for (c=0; c<colonne; c++)
{
    for (r=0; r<righe; r++)
        printf("%d ", trasposta[c][r]);
    printf("\n");
}

/* dealloco la matrice */
for (r=0; r<righe; r++)
    free(matrice[r]);
free(matrice);

/* dealloco la trasposta */
for (c=0; c<colonne; c++)
    free(trasposta[c]);
free(trasposta);
}

```



Esercitazione n. 5

Il file */tmp/circles.dat* contiene le coordinate del centro ed il raggio (x_C, y_C, r) di un numero fissato di cerchi definiti nel piano.

Scrivere un programma che, lette dal file tali informazioni, verifichi quanti cerchi sono tra loro parzialmente sovrapposti e riporti infine tale numero.

Il programma deve usare

- una struttura *Circle* che abbia come membri x, y, r
- una funzione *overlap* che accetta come argomenti due strutture *Circle* e ritorna 1 o 0 a seconda che i due cerchi corrispondenti siano parzialmente sovrapposti o meno
- una funzione *ReadData* che accetta come argomento il nome del file e ritorna un array di *Circle* allocato dinamicamente al suo interno .

Il file */tmp/circles.dat* contiene nella prima riga il numero di punti e nelle righe successive 4 numeri: un indice intero, le coordinate x_C e y_C , il raggio r .

circles.dat

```
20
0 -0.425 -0.214 0.5
1 -2.938 1.546 0.8
2 -1.363 0.336 1.3
3 0.764 -0.765 0.2
4 1.599 1.765 0.1
5 -1.364 1.346 0.6
6 0.417 0.932 0.2
7 -0.920 0.231 0.1
8 -0.826 -0.438 0.4
9 0.806 0.970 0.6
10 1.380 -1.498 0.1
11 1.768 1.230 0.2
12 1.386 0.651 0.8
13 0.050 1.989 1.0
14 1.152 0.130 0.3
15 -1.331 -0.938 1.1
16 0.403 -0.773 0.3
17 2.150 0.435 0.4
18 1.381 1.544 0.2
19 -0.649 -1.593 0.5
```

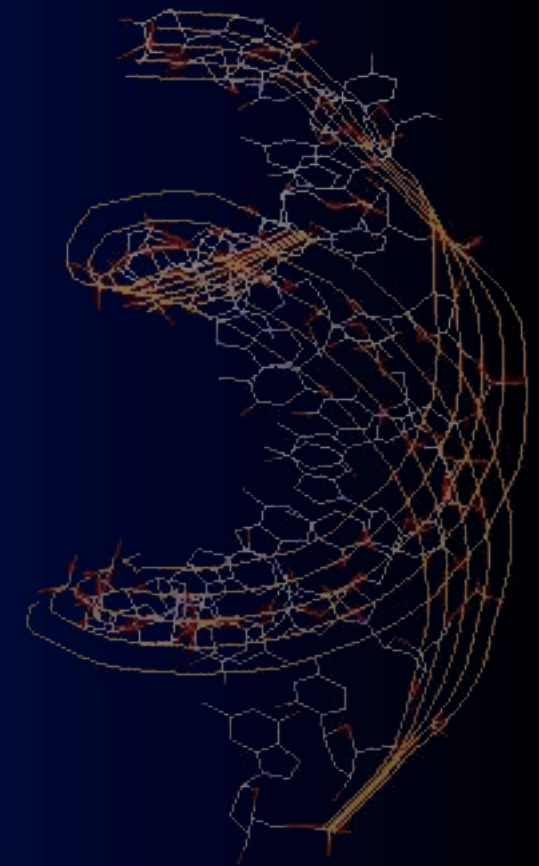


```

Circle *ReadData(char *file_dati,int *N)
{
    int a;
    int i;
    FILE *f;
    Circle *c;
    f = fopen(file_dati, "r+");
    fscanf(f,"%d",N);
    c = (Circle*)malloc((*N)*sizeof(Circle));
    for (i=0; i< *N; i++)
    {
        fscanf(f, "%d %f %f %f", &a, &c[i].x, &c[i].y, &c[i].r);
    }
    return c;
}

int overlap(Circle a, Circle b)
{
    int chk=0;
    float d,r2;
    r2=(a.r+b.r)*(a.r +b.r);
    d=(a.x - b.x)*(a.x -b.x)+(a.y - b.y)*(a.y - b.y);
    if(d <= r2) chk=1;
    return chk;
}

```



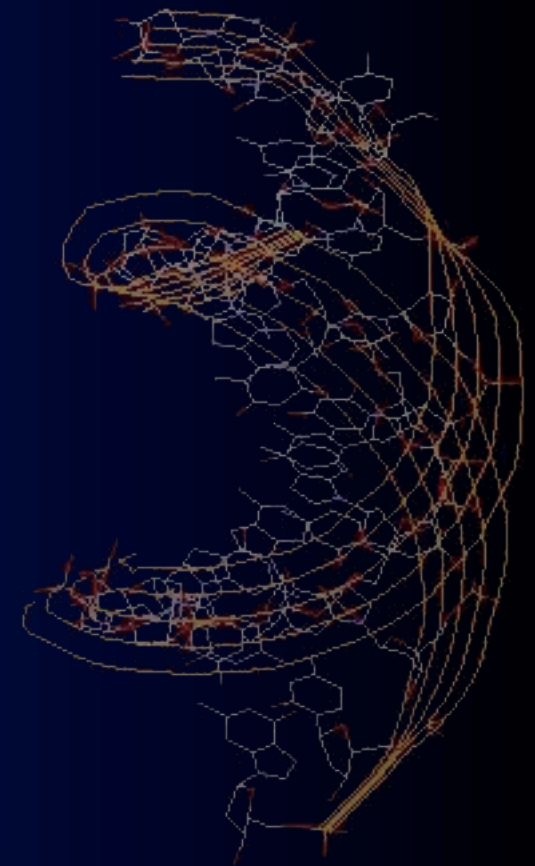
```

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
typedef struct circle
{
    float x;
    float y;
    float r;
} Circle ;

int overlap(Circle a, Circle b);
Circle *ReadData(char *file_dati, int *N);

int main(){
    char *file_dati;
    Circle *c;
    int i,j,N,count=0;
    file_dati="circles.dat";
    c=ReadData(file_dati,&N);
    for(i=0 ; i< (N-1) ; i++)
    {
        j=i+1;
        do{
            count += overlap(c[i],c[j]);
            j++;} while ( j < N);
    }
    printf(" numero sovrapposizioni %d \n", count);
    return 0;
}

```



Informazioni

- Le valutazioni delle prove pratiche saranno visionabili sulla pagina web del corso e su Webdocenti
- Il calendario degli esami (prova pratica) sarà disponibile sulla pagina web del corso e su Webdocenti
- Le date per gli orali saranno concordate con il docente
- E' necessario prenotarsi su Webdocenti per poter sostenere la prova pratica di esame