

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define NESSUNA 0
#define INTERSECA 1
#define CONTIENE 2
#define COINCIDE 3
#define SQR(x) (x)*(x)

struct cerchio{
    char nome[128];
    int x, y;
    unsigned int r;
};

struct cerchio *lettura_file(FILE *fp, int *n){
    int size=8;
    int n_conv;
    struct cerchio *vet;
    char buf[1000];

    *n=0;
    vet=malloc(sizeof(*vet)*size);
    while(fgets(buf, sizeof(buf), fp)){
        n_conv=sscanf(buf, "%s %d %d %d", vet[*n].nome,
&(vet[*n].x),
&(vet[*n].y), &(vet[*n].r));
        if(n_conv!=4)
            continue;

        (*n)++;

        if((*n)>=size){
            size*=2;
            vet=realloc(vet, sizeof(*vet)*size);
        }
    }
    vet=realloc(vet, sizeof(*vet)*(*n));
    return vet;
}

void stampa(struct cerchio *vet, int n){
    int i;
    for(i=0; i<n; i++)
        printf("%s %d %d %d\n", vet[i].nome, vet[i].x,
vet[i].y, vet[i].r);
}

int cmp_cerchi(const void *v1, const void *v2){
    const struct cerchio *c1=v1, *c2=v2;
    if(c1->r>c2->r)
        return 1;

```

```

        else if(c1->r<c2->r)
            return -1;
        else return 0;
        //return (c1->r)-(c2->r); alternativa a tutti gli if
    }

int relazioni(struct cerchio *c1, struct cerchio *c2){
    int rel=NESSUNA;
    float dist=sqrt(SQR(c1->x-c2->x)+SQR(c1->y-c2->y));
    if(dist<=(c1->r+c2->r)){
        if(c1->x==c2->x && c1->y==c2->y && c1->r==c2->r)
            rel=COINCIDE;
        else if(dist>=abs(c1->r-c2->r))
            rel=INTERSECA;
        else if(c1->r>c2->r)
            rel=CONTIENE;
    }
    return rel;
}

void stampa_relazioni(struct cerchio *vet, int n){
    char *rel[]={ "", "INTERSECA", "CONTIENE", "COINCIDE" };
    int i, j, k;
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            if(i!=j){
                k=relazioni(vet+i, vet+j);
                if(k!=NESSUNA)
                    printf("%s %s %s\n",
vet[i].nome, rel[k], vet[j].nome);
            }
        }
    }
}

int main(int argc, char *argv[]){

    if(argc!=2){
        fprintf(stderr, "#numero parametri sbagliati\n");
        return 1;
    }
    int n;
    struct cerchio *vet;
    FILE *fp;
    fp=fopen(argv[1], "r");
    if(fp==NULL){
        fprintf(stderr, "#errore nell'apertura del file\n");
        return 1;
    }
    vet=lettura_file(fp, &n);
    fclose(fp);

    printf("[CERCHI]\n");
    stampa(vet, n);
}

```

```
    qsort(vet, n, sizeof(*vet), cmp_cerchi);
    printf("[ORDINAMENTO]\n");
    stampa(vet, n);

    printf("[RELAZIONI]\n");
    stampa_relazioni(vet, n);

    free(vet);
    return 0;
}
```