

E9 – Esercizi sui puntatori in C

Esercizio 1. Scrivere una funzione per la risoluzione di equazioni di secondo grado (v. anche l'Esercizio 5 di E7) della forma $ax^2 + bx + c = 0$ in base alle seguenti indicazioni. La funzione riceve tre parametri (di tipo **double**) passati per valore che rappresentano i tre coefficienti a , b e c e due parametri $x1$ e $x2$ (anch'essi di tipo **double**) passati per riferimento. La funzione ritorna un valore intero.

Se l'equazione ha soluzioni reali la funzione ritorna 1 e nei parametri $x1$ e $x2$ vengono scritte le soluzioni dell'equazione. Se l'equazione non ha soluzioni reali la funzione ritorna 0 e le variabili $x1$ e $x2$ non vengono scritte.

Scrivere poi un programma che fa inserire all'utente i coefficienti di un'equazione di secondo grado e stampa le sue soluzioni se queste sono reali e un opportuno messaggio in caso contrario.

Esercizio 2. Modificare la funzione precedente in modo che calcoli le soluzioni complesse dell'equazione. Più precisamente, la funzione avrà tre parametri (di tipo **double**) passati per valore che rappresentano i coefficienti a , b e c dell'equazione e quattro parametri (di tipo **double**) passati per riferimento. Tali parametri rappresentano la parte reale e la parte immaginaria di ciascuna delle due soluzioni.

Modificare consistentemente il programma dell'esercizio precedente.

Esercizio 3. Scrivere una funzione che calcola l'equazione di una retta nel piano passante per due punti $(x1, y1)$ e $(x2, y2)$. La funzione riceve come parametri i quattro valori **double** $x1$, $y1$, $x2$ e $y2$ (passati per valore) e due ulteriori parametri m e q (passati per riferimento) in cui la funzione deve scrivere il coefficiente angolare m e il termine noto q dell'equazione della retta passante per i due punti dati. Qualora la retta sia verticale (cioè avente equazione $y=c$) il valore di m deve essere impostato a INFINITY (tale costante è definita nella libreria `math.h`) e q deve essere uguale a c .

Scrivere poi un programma che chiede all'utente di inserire le coordinate di due punti nel piano e stampa l'equazione della retta passante per i due punti dati.

Esercizio 4. Scrivere una funzione che prende come parametro un array di interi e restituisce il numero di valori positivi presenti nell'array. Si operi sugli elementi dell'array tramite l'aritmetica dei puntatori invece che tramite gli indici.

Scrivere poi un programma che usa la funzione scritta.

Esercizio 5. Scrivere una funzione che prende come parametro un array di interi e restituisce il numero di elementi che sono maggiori del loro successore. Si operi sugli elementi dell'array tramite l'aritmetica dei puntatori invece che tramite gli indici.

Scrivere poi un programma che usa la funzione scritta.

Soluzioni

Esercizio 1 - svolgimento.

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

int soluzioni(double a, double b, double c, double * x1, double * x2);

int main(void) {

    printf("Inserisci i tre coefficienti dell'equazione\n");
    double a, b, c;
    scanf("%lf%lf%lf", &a, &b, &c);
    double x1, x2;
    if(soluzioni(a, b, c, &x1, &x2))
        printf("Le soluzioni reali dell'equazione sono: %g e %g\n",
x1, x2);
    else
        printf("L'equazione non ha soluzioni reali\n");

}

int soluzioni(double a, double b, double c, double * x1, double * x2) {
    double delta=b*b-4*a*c;
    int soluzioniReali=0;
    if(delta>0) {
        *x1=(-b+sqrt(delta))/(2*a);
        *x2=(-b-sqrt(delta))/(2*a);
        soluzioniReali=1;
    }

    return soluzioniReali;
}
```

Esercizio 2 - svolgimento.

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

void soluzioni(double a, double b, double c, double * re1, double * im1,
double * re2, double * im2);

int main(void) {

    printf("Inserisci i tre coefficienti dell'equazione\n");
    double a, b, c;
    scanf("%lf%lf%lf", &a, &b, &c);
    double re1, im1, re2, im2;
    soluzioni(a, b, c, &re1, &im1, &re2, &im2);
    printf("Le soluzioni dell'equazione sono: %g + i %g e %g + i %g\n",
re1, im1, re2, im2);
}

void soluzioni(double a, double b, double c, double * re1, double * im1,
double * re2, double * im2) {
    double delta=b*b-4*a*c;
    if(delta>0) {
```

```

        *re1=(-b+sqrt(delta))/(2*a);
        *re2=(-b-sqrt(delta))/(2*a);
        *im1=0;
        *im2=0;
    }else{
        *re1=-b/(2*a);
        *im1=sqrt(-delta)/(2*a);
        *re2=*re1;
        *im2=-sqrt(-delta)/(2*a);
    }
}

```

Esercizio 3 - svolgimento.

```

#include <stdio.h>
#include <math.h>

```

```

void retta(double x1, double y1, double x2, double y2, double * m, double
* q);

```

```

int main(void) {

```

```

    printf("Inserisci le coordinate di un punto nel piano\n");
    double x1, y1;
    scanf("%lf%lf",&x1, &y1);

```

```

    printf("Inserisci le coordinate di un secondo punto\n");
    double x2, y2;
    scanf("%lf%lf",&x2, &y2);

```

```

    double m, q;
    retta(x1, y1, x2, y2, &m, &q);

```

```

    printf("L'equazione della retta per i due punti dati e'\n");
    if(m==INFINITY)
        printf("x=%g\n",q);
    else if(m==0)
        printf("y=%g\n",q);
    else
        printf("y=%g x + %g\n", m, q);

```

```

}

```

```

void retta(double x1, double y1, double x2, double y2, double * m, double
* q) {

```

```

    if(x1==x2){ //retta verticale
        *m=INFINITY;
        *q=x1;
    }else{
        *m=(y2-y1)/(x2-x1);
        *q=-(*m)*x2+y2;
    }

```

```

}

```

Esercizio 4 - svolgimento.

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

int contaPositivi(int* aPtr, int dim);

int main(void) {

    printf("Quanti elementi vuoi inserire?\n");
    int n;
    scanf("%d", &n);
    int a[n];
    for(int i=0; i<n; i++){
        printf("Inserisci l'elemnto in posizione %d ", i);
        scanf("%d", &a[i]);
    }
    int c=contaPositivi(a, n);
    printf("Il numero di elementi positivi inseriti e': %d\n", c);
}

int contaPositivi(int* aPtr, int dim){
    int count=0;
    for(int i=0; i<dim; i++){
        if(*(aPtr+i)>0)
            count++;
    }
    return count;
}
```

Esercizio 5 - svolgimento.

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

int contaInversioni(int* aPtr, int dim);

int main(void) {

    printf("Quanti elementi vuoi inserire?\n");
    int n;
    scanf("%d", &n);
    int a[n];
    for(int i=0; i<n; i++){
        printf("Inserisci l'elemnto in posizione %d ", i);
        scanf("%d", &a[i]);
    }
    int c=contaInversioni(a, n);
    printf("Il numero di elementi maggiore del proprio successore e':
%d\n", c);
}

int contaInversioni(int* aPtr, int dim){
    int count=0;
    for(int i=0; i<dim-1; i++){
        if(*(aPtr+i)>*(aPtr+i+1))
            count++;
    }
    return count;
}
```