

25-11-2002

## Compito in classe

Risolvere i seguenti problemi:

Alberare un viale.

**Ob.** Determinare il numero di piante

**Ris.** Piante

**Dati.** Lunghezza distanza

**Limiti.** Lunghezza > 0 distanza > 0

**Algoritmo.**

Leggi lunghezza, distanza

Mentre (lunghezza < 0) or (distanza < 0)

    Leggi lunghezza, distanza

    Piante = lunghezza / distanza

    Scrivi piante

**Codifica**

```
main() {  
    float lunghezza, distanza;  
    int piante;  
    scanf("%f %f", &lunghezza, &distanza);  
    while ((lunghezza < 0) || (distanza < 0))  
        scanf("%f %f", &lunghezza, &distanza);  
    piante = lunghezza / distanza;  
    printf("%d\n", piante);  
}
```

Una piazza è pavimentata con n piastrelle di forma quadrata di lato x.  
Calcolare l'area della piazza.

**Ob.** Determinare l'area

**Ris.** area

**Dati.** Lato, piastrelle

**Limiti.** lato > 0 piastrelle > 0

**Algoritmo.**

Leggi lato, piastrelle

Mentre (lato < 0) or (piastrelle < 0)

    Leggi lato, distanza

Area = piastrelle \* lato \* lato;

Scrivi area

**Codifica.**

```
main() {  
    float lato, area;  
    int piastrelle;  
    scanf("%f %d", &lato, &piastrelle);  
    while ((lato < 0) || (piastrelle < 0))  
        scanf("%f %d", &lato, &piastrelle);  
    area = piastrelle * lato * lato;  
    printf("%f\n", area);  
}
```

Dati n contenitori di forma cilindrica determinare quanti di questi possono contenere una determinata quantità di acqua.

**Ob.** Determinare il numero di contenitori

**Ris.** contenitori

**Dati.** N, litriContenitore, acqua

**Limiti.**  $N > 0$ , litriContenitore, litriAcqua

**Algoritmo.**

Leggi n, litriAcqua

Mentre  $(n < 0)$  or  $(\text{litriAcqua} < 0)$

    Leggi n, litriAcqua

Per n volte

    Leggi litriContenitore

    Se  $\text{litriContenitore} > \text{litriAcqua}$

        Contenitori = contenitori + 1

Scrivi contenitori

**Codifica.**

```
main() {
    float litriAcqua, litriContenitore;
    int i, n, contenitori = 0;
    scanf("%f %d", &litriAcqua, &n);
    while ((litriAcqua < 0) || (n < 0))
        scanf("%f %d", &litriAcqua, &n);
    for(i=0; i<n; ++i) {
        scanf("%f", &litriContenitore);
        if( litriContenitore > litriAcqua ) contenitori++;
    }
    printf("%d\n", contenitori);
}
```

Sommare i primi n numeri interi positivi divisibili per 3.

**Ob.** Determinare la somma dei primi n numeri

**Ris.** somma

**Dati.** N

**Limiti.**  $N > 0$

**Algoritmo.**

Leggi n

Mentre ( $n < 0$ )

    Leggi n

Somma = 0;

numero = 0;

Per n volte

    Numero = numero + 3

    Somma = somma + numero

Scrivi somma

**Codifica.**

```
main() {
    int i, n, somma =0, numero =0;
    scanf(" %d", &n);
    while ( (n < 0)
        scanf("%d", &n);
    for(i=0; i<n; ++i) {
        numero = numero + 3;
        somma = somma + numero;
    }
    printf("%d\n", somma);
}
```

Dati n numeri determinare il massimo.

**Ob.** Determinare il massimo tra n numeri

**Ris.** massimo

**Dati.** N, numero

**Limiti.**  $N > 0$

**Algoritmo.**

```
Leggi n
Mentre (n < 0)
    Leggi n
Leggi massimo
Per n - 1 volte
    Leggi numero
    Se numero > massimo
        Massimo = numero
Scrivi massimo
```

**Codifica.**

```
main() {
    int i, n;
    float massimo, numero;
    scanf("%d", &n);
    while ( (n < 0)
        scanf("%d", &n);
    scanf("%f", &massimo);
    for(i=1; i<n; ++i) {
        scanf("%f", &numero);
        if( numero > massimo ) massimo = numero;
    }
    printf("%f\n", massimo);
}
```

Leggere una successione crescente di numeri.

**Ob.** Determinare se una successione di numeri è crescente

**Ris.** “è crescente” “non è crescente”

**Dati.** N, numero

**Limiti.**  $N > 0$

**Algoritmo.**

```
Leggi n
Mentre (n < 0)
    Leggi n
Mentre (n > 0) and crescente
    Leggi numero
    Crescente = numero > precedente
    Precedente = numero
    n--
Se crescente   Scrivi “è crescente”
Altrimenti    Scrivi “non è crescente”
```

**Codifica.**

```
main() {
    int n, crescente = 1;
    float numero, precedente;
    scanf(“ %d”, &n);
    while ( (n < 0)
        scanf(“%d”, &n);
    scanf(“%f”, &precedente);
    scanf(“%f”, &massimo);
    while ( (n > 0) && crescente) {
        scanf(“%f”, &numero);
        crescente = numero > precedente;
        precedente = numero;
        n--;
    }
    if( crescente ) printf(“è crescente\n”);
    else printf(“non è crescente\n”);
}
```

