

# SOLUZIONE 1

---

Scrivere un programma C che definisca la struttura “misuraUK” composta da quattro interi (miglia,yarde,piedi,pollici), e una struttura “misuraUE” composta da tre interi (km,metri,mm). Si richieda poi una misura inglese e si stampi a video la misura europea.

```
typedef struct { int miglia;int yarde;int piedi;int pollici;} misuraUK ;

typedef struct { int km;int metri;int mm;} misuraUE;

main(){

    misuraUK length;
    misuraUE lung;
    double totpollici,totmm;

    scanf("%d %d %d %d",&length.miglia,&length.yarde,&length.piedi,&length.pollici);

    totpollici=length.pollici+12*(length.piedi+3*(length.yarde+1760*length.miglia));
    totmm=totpollici*25.4;

    lung.km=(int)totmm/1000000;
    lung.metri=((int)totmm/1000)%1000;
    lung.mm=(int)totmm%1000;

    printf("\n%d km %d m %d mm\n",lung.km, lung.metri,lung.mm);
}
```

# SOLUZIONE 2

---

Scrivere un programma C che definisca la struttura “persona” composta da nome, cognome, indirizzo, provincia e data di nascita. Si richiedano tutti i dati e si stampino a video.

```
typedef struct {
    int giorno;
    int mese;
    int anno;
} data ;

typedef struct {
    char nome[40];
    char cognome[40];
    struct data nascita;
    char indirizzo[100];
    char provincia[2];
} persona ;

void main(){

    persona p;

    // continua...
```

*Continua...*

# SOLUZIONE 2

---

```
//...continua
```

```
printf("Inserisci il nome: ");  
scanf("\n%s", p.nome);
```

```
printf("\nInserisci il cognome: ");  
scanf("\n%s", p.cognome);
```

```
printf("\nInserisci la data di nascita (gg/mm/aaaa): ");  
scanf("\n%d/%d/%d", &p.nascita.giorno, &p.nascita.mese, &p.nascita.anno);
```

```
fflush(stdin);  
printf("\nInserisci l'indirizzo: ");  
gets(p.indirizzo);
```

```
printf("\nInserisci la provincia (2 caratteri): ");  
scanf("\n%s", p.provincia);
```

```
printf("\n%s %s\n", p.nome, p.cognome);  
printf("\n%s\t(%s)", p.indirizzo, p.provincia);
```

```
}
```

# SOLUZIONE 3

Si scriva un programma che richieda in ingresso un numero di elementi N (<20) e poi N interi, inserendoli in un array. Chieda poi una soglia S e si copino tutti gli elementi del vettore in altri due vettori, separando elementi sopra la soglia e sotto la soglia.

```
#include <stdio.h>
#define DIM 20

void main () {

    float soglia, vett[DIM], sopra[DIM], sotto[DIM];
    int max, i, j=0, k=0;

    printf("Quanti valori vuoi inserire [< %d] ? ", DIM);
    scanf("\n%d", &max);
    while (max >= DIM) {
        printf("\nErrore. Il valore deve essere < %d.\nInserisci di nuovo: ", DIM);
        scanf("\n%d", &max);
    }

    for(i = 0; i < max; i++) {
        printf("\nInserisci il valore %d : ", (i+1));
        scanf("\n%f", &vett[i]);
    }

    printf("\nInserisci la soglia : ");
    scanf("\n%f", &soglia);
```

*Continua...*

# SOLUZIONE 3

---

```
// ...continua
```

```
for(i = 0; i < max; i++) {
    if (vett[i] > soglia) {
        sopra[j] = vett[i];
        j++;
    }
    else if (vett[i] < soglia) {
        sotto[k] = vett[i];
        k++;
    }
}

printf("\nVettore inserito : ");
for(i = 0; i < max; i++) printf("  %.2f", vett[i]);
printf("\n\nVettore sopra-soglia : ");
for(i = 0; i < j; i++) printf("  %.2f", sopra[i]);
printf("\nCi sono %d elementi sopra la soglia.\n", j);
printf("\nVettore sotto-soglia : ");
for(i = 0; i < k; i++) printf("  %.2f", sotto[i]);
printf("\nCi sono %d elementi sotto la soglia.\n", k);
}
```

# SOLUZIONE 4

---

Si scriva un programma C che chieda in ingresso due matrici 3x3 e poi chieda all'utente cosa vuole fare: somma, sottrazione o moltiplicazione. Si stampi a video la matrice risultato.

```
#include <stdio.h>
#define DIM 3
#define TRUE 1
#define FALSE 0
typedef int boolean;

void main() {
    int matr1[DIM][DIM];
    int matr2[DIM][DIM];
    int result[DIM][DIM];
    int i,r,rr,c,elemento;
    boolean operazione;

    printf("Matrice 1:\n");
    for(r=0; r<DIM; r++) {
        printf("Inserisci la %d riga di %d elementi: ",(r+1),DIM);
        scanf("%d %d %d",&matr1[r][0],&matr1[r][1],&matr1[r][2]);}

    printf("\nMatrice 2:\n");
    for(r=0; r<DIM; r++) {
        printf("Inserisci la %d riga di %d elementi: ",(r+1),DIM);
        scanf("%d %d %d",&matr2[r][0],&matr2[r][1],&matr2[r][2]);}
```

*Continua...*

# SOLUZIONE 4

---

```
printf("\n\nScegli un'operazione: \n");
printf("\n1. Somma delle matrici.\n");
printf("2. Differenza delle matrici.\n");
printf("3. Prodotto delle matrici.\n");
printf("4. Uscita.\n");
printf("\nScelta: ");
scanf("\n%d",&i);

if(i == 1) { operazione = TRUE;
    for(r=0;r<DIM;r++) {
        for(c=0;c<DIM; c++) {
            result[r][c] = matr1[r][c] + matr2[r][c];
        }
    }
}
else if(i == 2) {operazione = TRUE;
    for(r=0;r<DIM;r++) {
        for(c=0;c<DIM; c++) {
            result[r][c] = matr1[r][c] - matr2[r][c];
        }
    }
}
else if(i == 3) {operazione = TRUE;
    for(r=0;r<DIM;r++) {
        for(c=0;c<DIM;c++) { elemento = 0;
            for(rr=0; rr<DIM; rr++) {
                elemento = elemento + matr1[r][rr]*matr2[rr][c];
            }
            result[r][c] = elemento;
        }
    }
}
```

*Continua...*

# SOLUZIONE 4

---

```
else if(i == 4) {
    operazione = FALSE;
    printf("\nUscita.\n");
}
else {
    operazione = FALSE;
    printf("\nERRORE: Scelta non supportata: %d\n",i);
}

// Stampo la matrice:
if(operazione) {
    printf("\nMatrice risultato: \n");
    for(r=0;r<DIM;r++) {
        for(c=0;c<DIM;c++) {
            printf("%d\t",result[r][c]);
        }
        printf("\n");
    }
}
```



# SOLUZIONE 5

---

Si scriva un programma che richieda in ingresso l'andamento annuale del valore delle azioni (12 valori, uno per ogni mese) e crei un vettore. Poi stampi a video un istogramma orizzontale.

```
#include <stdio.h>
#define MAXSTELLE 50
#define MESI 12

void main(){
    int valore[MESI];
    int i, j, max=0, stelle;
    printf("Inserisci le quotazioni mensili delle azioni\n");
    for (i=0; i<MESI; i++) {
        printf("Mese %d: ", i+1);
        scanf("\n%d", &valore[i]);
        if (valore[i] > max) max = valore[i];
    }

    printf("\nmese\tandamento\n");
    for (i=0; i<MESI; i++) {
        printf("\n %d\t", i+1);
        stelle = (int)(valore[i] * MAXSTELLE) / max;
        for (j=0; j<stelle; j++) printf("*");
        printf(" %d", valore[i]);
    }
    printf("\n\n");
}
```

# SOLUZIONE 6

---

Si realizzi un programma C che legga da utente i dati relativi ad alcuni corsi. In particolare, per ogni corso vengono dati: **denominazione del corso**: **cognome del docente**: **iscritti**.

Il programma deve stampare la denominazione del corso e il cognome del docente relativi a tutti i corsi che hanno il numero di iscritti maggiore o uguale alla media aritmetica degli iscritti (calcolata su tutti i corsi).

```
#include <stdio.h>
#define N 30
typedef struct {
    char denominazione[20];
    char cognome_docente[15];
    int studenti;
} stud ;

void main()
{   int i, nc;
    float somma media;
    stud corsi[N];
    printf("Inserisci il numero dei corsi ");
    scanf("%d", &nc);
```

*Continua...*

# SOLUZIONE 6

---

```
/* inserimento dati */
for (i=0; i<nc; i++)
{   printf("Inserisci il nome del corso ");
    scanf("%s",corsi[i].denominazione);
    printf("Inserisci il cognome del docente ");
    scanf("%s",corsi[i].cognome_docente);
    printf("Inserisci il numero degli iscritti");
    scanf("%d",&corsi[i].studenti);
}

somma=0;
for (i=0; i< nc; i++)
somma=somma + corsi[i].studenti;
media= somma/nc;
for (i=0; i< nc; i++)
    if (corsi[i].studenti>=media)
        printf("%s %s\n",corsi[i].denominazione,
                corsi[i].cognome_docente);
}
```

# SOLUZIONE 7

---

- Si scriva un programma C che legga una serie di dati e li memorizzi in un vettore SQUADRE (di dimensione 3) contenente strutture (**struct squadra**) del tipo:
  - **nome squadra** (*stringa di lunghezza 20*)
  - **codice squadra** (*intero*)
  - **goal fatti** (*intero*)
  - **goal subiti** (*intero*)
- Stampi a terminale tutti i nomi e codici delle squadre che hanno fatto un numero di goal maggiore del numero dei goal subiti.
- Letto a terminale un codice di una squadra stampi a video il nome della squadra, i goal fatti e i goal subiti.

# SOLUZIONE 7

---

```
#include <stdio.h>
#define N 30
typedef struct {
    char nome[20];
    int codice;
    int goal_fatti, goal_subiti;
} squadra;

void main()
{
    int i, ns, cod, T;
    squadra SQUADRE[N];
    printf("Inserisci il numero delle squadre");
    scanf("%d", &ns);
    /* inserimento dati */
    for (i=0; i<ns; i++)
    {
        printf("Inserisci nome, codice, goal fatti e subiti \n");
        scanf("%s", SQUADRE[i].nome);
        scanf("%d", &SQUADRE[i].codice);
        scanf("%d", &SQUADRE[i].goal_fatti);
        scanf("%d", &SQUADRE[i].goal_subiti);
    }
}
```

*Continua...*

# SOLUZIONE 7

---

```
/* punto 2 */
for (i=0; i<ns; i++)
    { if(SQUADRE[i].goal_fatti> SQUADRE[i].goal_subiti)
        printf("%s\n", SQUADRE[i].nome);
        printf("%d\n", SQUADRE[i].codice);
    }

/* punto 3 */
printf("Inserisci un codice ");
scanf("%d", &cod);
i=0; T=0;
while ((i < ns)&& (T==0))
    { if (SQUADRE[i].codice == cod)
        {printf("%s\n", SQUADRE[i].nome);
            printf("%d\n", SQUADRE[i].goal_fatti);
            printf("%d\n", SQUADRE[i].goal_subiti);
            T=1;}
        i++;
    }
if (T==0) printf("codice non trovato");
}
```

# SOLUZIONE 8

---

Scrivere un programma C che crei una rubrica telefonica, usando una struct “persona” (nome, cognome, numero di telefono, e-mail). Il programma chiede di inserire delle nuove persone, poi si mette in modalità ricerca e, chiesto il nome stampa a video la scheda corrispondente.

```
#include <stdio.h>
#include <string.h>
#define DIM 20

//struttura dati
typedef struct {
    char nome[40];
    char cognome[40];
    char numero[13];
    char email[100];
} persona;

void main(){

    persona rubrica[DIM];
    char scelta = 's';
    int indice = 0;
    int indric = 0;
    char nomic[40];
    int num = 0;
```

*Continua...*

# SOLUZIONE 8

---

```
//...continua
//inserimento dati
printf("Inserisci un contatto: \n");
// Ciclo di inserimento:
while((scelta == 'S' || scelta == 's') && indice < DIM) {
    printf("\nInserisci il nome: ");
    scanf("\n%s", rubrica[indice].nome);

    printf("Inserisci il cognome: ");
    scanf("\n%s", rubrica[indice].cognome);

    printf("Inserisci il numero di telefono: ");
    scanf("\n%s", rubrica[indice].numero);

    printf("Inserisci l'e-mail: ");
    scanf("\n%s", rubrica[indice].email);

    indice++;

    // Chiedo se continuare o meno.
    printf("\nSpazio disponibile: %d\n", (DIM-indice));
    printf("Inserire un altro contatto? [S/N]: ");
    scanf("\n%c",&scelta);
}
if(indice == DIM) {
    printf("\n\nRubrica piena.\n\n");
}
```

*Continua...*



# SOLUZIONE 8

---

```
//...continua
    scelta = 'S';
    // Effettuiamo la ricerca:
    while(scelta == 'S' || scelta == 's') {
        num = 0;
        indric = 0;

        printf("\n\nInserisci il nome da ricercare: ");
        scanf("\n%s", nomic);

        while(indric < DIM) {
            if(!strcmp(nomic, rubrica[indric].nome)) {
                printf("\n\n_____ \n\n");
                printf("%s ", rubrica[indric].nome);
                printf("%s\n", rubrica[indric].cognome);
                printf("Tel: %s\n", rubrica[indric].numero);
                printf("Email: %s\n", rubrica[indric].email);
                printf("_____ \n");
                num++;
            }
            indric++;
        }
        printf("\n\nNumero occorrenze: %d\n", num);

        printf("\nVuoi fare un'altra ricerca? [S/N]: ");
        scanf("\n%c",&scelta);
    }
}
```