

Scrivere un menu che permetta:

- ✓ l'inserimento;
- ✓ la visualizzazione;
- ✓ l'eliminazione;
- ✓ l'ordinamento;

di interi da un elenco di massimo 10 elementi, utilizzando puntatori e strutture.

```
#include <stdio.h>
#define MAX 10

typedef enum {VUOTA, PIENA} dispon;

typedef struct {
    int elemento;
    dispon stato;
} cella;

cella vett[MAX];
cella *p, *s, *m;

int operaz=-1, numero;
```

```
void main()
{
    for (p=vett;p<(vett + MAX);p++)
        p->stato=VUOTA;

    while (operaz!=0)
    {

        /* richiesta operazione da eseguire */
        printf("\nInserisci l'operazione (0 fine,
                1 aggiungi, 2 elimina, 3 visualizza,
                4 ordina): ");

        scanf("%d",&operaz);
    }
}
```

```
switch (operaz)
{
    case 0:
        printf("\nFine Programma!");
        break;
    case 1: //inserimento
        break;
    case 2: //eliminazione
        break;
    case 3: // visualizzazione array
        break;
    case 4: // ordinamento in base a elemento
        break;
    default:
        printf("\nOperazione non supportata!");
        break;
}
```

```
case 1: //inserimento
        /* richiesta numero sul quale operare */

        printf("\nInserisci il numero: ");
        scanf("%d",&numero);
        /* ricerca cella disponibile */
        i=0;
        while(i<MAX && vett[i].stato==PIENA)
            i++;
        /* verifica inserimento */
        if (i==MAX)
            printf("\nNon ci sono celle disponibili!");
        else
        {
            vett[i].elemento=numero;
            vett[i].stato=PIENA;
        }

        break;
```

```
case 2: //eliminazione
/* richiesta numero sul quale operare */

printf("\nInserisci il numero positivo: ");
scanf("%d",&numero);
/* ricerca numero nell'array */
i=0;
while(i<MAX && (vett[i].stato==VUOTA) ||
      (vett[i].stato==PIENA
       && (vett[i].elemento!=numero)))
    i++;

if (i==MAX)
    printf("\nNumero non presente!");
else
    vett[i].stato=VUOTA;

break;
```

case 3: // visualizzazione array

```
for(p=vett;p<(vett + MAX);p++)  
{  
    if (p->stato==PIENA)  
        printf("\n vettore %d-esimo (PIENA) = %d",  
                p,          p->elemento);  
  
    else  
        printf("\n vettore %d-esimo (VUOTA)", p);  
}
```

break;

case 4: // ordinamento in base a elemento

```
for (p=vett;p<(vett + MAX -1);p++)
{
    if (p->stato==PIENA)
    {
        m=p;
        for(s=p+1;s<(vett + MAX);s++)
            if ((s->stato==PIENA) &&(m->elemento>s->elemento))
                m=s;
        if (m!=p) {
            numero=m->elemento;
            m->elemento=p->elemento;
            p->elemento=numero;
        }
    }
}
```

break;



Esempio di utilizzo dei  
puntatori per la selezione  
delle variabili su cui operare.

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

```
int a, b, c;
```

```
int *p;
```

```
char risp;
```

```
void main()
```

```
{
```

```
    /* inizializzazione */
```

```
    a = 5;
```

```
    b = 8;
```

```
    c = 3;
```

```
printf("\na = %d, b = %d, c = %d", a, b, c);  
  
do {  
    printf("\nQuale variabile vuoi  
        raddoppiare (f per finire)? ");  
  
    fflush(stdin);  
  
    scanf("%c", &risp);
```

```
switch(risp)
{
    case 'a':          p = &a;
                        break;
    case 'b':          p = &b;
                        break;
    case 'c':          p = &c;
                        break;
    case 'f':          printf("\nFine");
                        p = NULL;
                        break;
    default:
                        printf("\nVariabile non valida");
                        p = NULL;
}
```

```
if (p != NULL)
    *p = *p * 2;

printf("\na = %d, b = %d, c = %d"
      , a,      b,      c);

} while (resp != 'f');
}
```

Scrivere un programma che inizializza un vettore di 10 interi con valori casuali e lo visualizza.

Successivamente il programma chiede ripetutamente all'utente quali valori scambiare (basandosi sull'indice), li scambia facendo uso dei puntatori, e stampa a video il vettore risultante.

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

int vett[10];
int *p1, *p2;

int el1, el2, i, tmp;

void main()
{
    /* inizializzazione */
    for (i=0; i<10; i++) {
        vett[i]= rand() % 20;
        printf("%d\t", vett[i]);
    }
```

```
do {  
    printf("\nPrimo elemento da scambiare  
          (-1 per finire)? ");  
    scanf("%d", &el1);  
    printf("\nSecondo elemento da scambiare  
          (-1 per finire)? ");  
    scanf("%d", &el2);  
  
    if ( (el1 >= 0 ) && (el1 < 10) )  
        p1 = &vett[el1];  
    else p1=NULL;  
    if ( (el2 >= 0 ) && (el2 < 10) )  
        p2 = &vett[el2];  
    else p2=NULL;  
  
    // il do continua...
```



```
if ( (p1 != NULL) && (p2 != NULL) ) {  
    tmp = *p1;  
    *p1 = *p2;  
    *p2 = tmp;  
}  
//fine if  
  
/* visualizzazione vettore */  
printf("\n");  
for (i=0; i<10; i++)  
    printf("%d\t", vett[i]);  
  
} while ( (el1 != -1) && (el2 != -1) );  
  
}
```

Gestione di un magazzino contenente articoli (codice, descrizione, quantità, prezzo). Permettere l'inserimento, la modifica della quantità di un articolo ricercato mediante il codice e la visualizzazione dell'archivio risultante.

## Esercizio P\_6 - Magazzino

36

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 3

typedef struct {
    char    codice[10];
    char    descrizione[100];
    int     qta;
    int     prezzo;
} articolo;

articolo magazzino[MAX], tmp;
articolo *p;
int j, risp, qta;
char codice[10];
```

```
void main() {                                     /* inserimento articoli */
    p = &tmp;
    for(j=0; j<MAX; j++) {
        printf("\nArticolo %d: ", j);
        printf("\n\t codice: ");
        scanf("%s", p->codice);
        printf("\n\t descrizione: ");
        scanf("%s", p->descrizione);
        printf("\n\t quantita': ");
        scanf("%d", &(p->qta) );
        printf("\n\t prezzo: ");
        scanf("%d", &(p->prezzo) );
        /* copia dalla variabile tmp all'elemento
           dell'array */
        magazzino[j]=*p;
    }
```

```
/* aggiornamento quantita'*/  
printf("\nInserisci articolo da aggiornare (codice): ");  
scanf("%s", codice);  
printf("\nInserisci quantita' venduta: ");  
scanf("%d", &qta);  
  
p = NULL;  
for(j=0; j<MAX; j++)  
    if (strcmp(codice, magazzino[j].codice) == 0 )  
        p = &magazzino[j];  
  
if (p != NULL)  
    p->qta = p->qta - qta;
```

```
/* visualizzazione risultato */  
printf("\n\n");  
for(j=0; j<MAX; j++) {  
    /* p punta direttamente all'elemento  
    del vettore */  
    p = &magazzino[j];  
  
    printf("\nArticolo %d: ", j);  
    printf("\n\t codice: %s", p->codice);  
    printf("\n\t descrizione: %s", p->descrizione);  
    printf("\n\t quantita': %d", p->qta);  
    printf("\n\t prezzo: %d", p->prezzo);  
}  
}
```

Esempi di utilizzo di puntatori  
semplici e doppi.  
Dire cosa viene visualizzato a  
schermo.

```
#include <stdio.h>
int a, b, c;
int *x, *y;
int **z;
void main()
{
    /* inizializzazione */
    a = 5;
    b = 8;
    c = 3;
    printf("\na = %d, b = %d, c = %d", a, b, c);
    /* stampa : 5 8 3 */

    x = &a;
    y = &b;
    printf("\n*x = %d, *y = %d", *x, *y);
    /* stampa 5 8 */
}
```



```
z = &x;
printf("\n**z = %d", **z);
/* stampa 5 */

a = 34; /* modifica dei dati */
*x = 53;
z = &y;
y = &c;
**z = 11;
printf("\na = %d, b = %d, c = %d", a, b, c);
/* stampa : 53 8 11 */
printf("\n*x = %d, *y = %d", *x, *y);
/* stampa 53 11 */
printf("\n**z = %d", **z);
/* stampa 11 */
}
```

Esempio di uso di puntatori  
con le stringhe.

```
main()
{
    char name[5];
    name[0] = 'A';
    name[1] = 'n';
    name[2] = 'n';
    name[3] = 'a';
    name[4] = '\0';

    printf("Il nome e' %s\n", name);
    /* stampa : Il nome e' Anna */
    printf("Una lettera e' %c\n", name[2]);
    /* stampa : Una lettera e' n */
    printf("Parte del nome e' %s\n", &name[1]);
    /* stampa : Parte del nome e' nna */
}
```

Esempi di scambio di variabili  
senza e con puntatori.

```
#include <stdio.h>
main()
{
    int x, y, z;

    printf("\nPrimo valore (x): ");
    scanf("%d",&x);

    printf("\nSecondo valore (y): ");
    scanf("%d",&y);

    z=x;
    x=y;
    y=z;
    printf("\nValori scambiati x=%d, y=%d",x,y);
}
```

## Esercizio P\_9 - Variabili

47

```
#include <stdio.h>
main()
{
    int x, y, z;
    int *px, *py, *pz;

    px=&x;  py=&y;  pz=&z;

    printf("\nPrimo valore (x): ");
    scanf("%d",px);
    printf("\nSecondo valore (y): ");
    scanf("%d",py);

    pz=px;
    px=py;
    py=pz;
    printf("\nValori scambiati x=%d, y=%d",*px,*py);
}
```

Esempio di utilizzo dei  
puntatori con gli array.  
Leggere il codice e capire  
cosa viene stampato.

## Esercizio P\_10 - Esempi d'uso

49

```
#include <stdio.h>
int vet[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
int *p, i;
void main() {
    p = vet;
    printf("(p = vet) --> %d \n", *p);
    p = &vet[3];
    printf("(p = &vet[3]) --> %d \n", *p);
    printf("\n");
    p = p + 5;
    *p = 43;

    p = vet + 3;
    *p = 99;

    p--;
    *p = *(p + 4);
    p = vet;
    for (i=0; i<10; i++)
        printf("(p+%d) --> %d \n", i, *(p+i));
}

/* stampa: (p = vet) --> 0 */
/* stampa: (p = &vet[3]) --> 3 */
/* stampa:
*(p+0) --> 0
*(p+1) --> 1
*(p+2) --> 6
*(p+3) --> 99
*(p+4) --> 4
*(p+5) --> 5
*(p+6) --> 6
*(p+7) --> 7
*(p+8) --> 43
*(p+9) --> 9
*/
```



Esempio di utilizzo dei  
puntatori con gli array.  
Leggere il codice e capire  
cosa viene stampato.

## Esercizio P\_11 - Esempi d'uso 2

51

```
#include <stdio.h>
#define dimM 10
int vet[dimM]={100,101,102,103,104,105,106,107,108,109};
int A=6, i;
int *p[5], *V;
void main() {
    for (i=0; i<5;i++)
        p[i]=&A;
    V = vet + 1;
    p[3]=&V[5];
    p[2]=&V[A];
    p[4]=V-1;
    for (i=0; i<4; i++)
        *p[i+1]=i+5;
    for (i=0; i<dimM - 1; i++)
        printf("%d - ", *(V+i));
    printf("\n");
    for (i=0; i<5; i++)
        printf("%d - ", **(p+i));
    printf("\n");
    for (i=0; i<dimM; i++)
        printf("%d - ", vet[i]);
    printf("\n%d", A);
}
```

/\* stampa:  
101 - 102 - 103 - 104 - 105 - 7 -  
6 - 108 - 109 - 5 - 5 - 6 - 7 -  
8 - 8 - 101 - 102 - 103 - 104 -  
105 - 7 - 6 - 108 - 109 - 5  
\*/