ESERCIZI DI PROGRAMMAZIONE DA TEMI D'ESAME

- vettori -

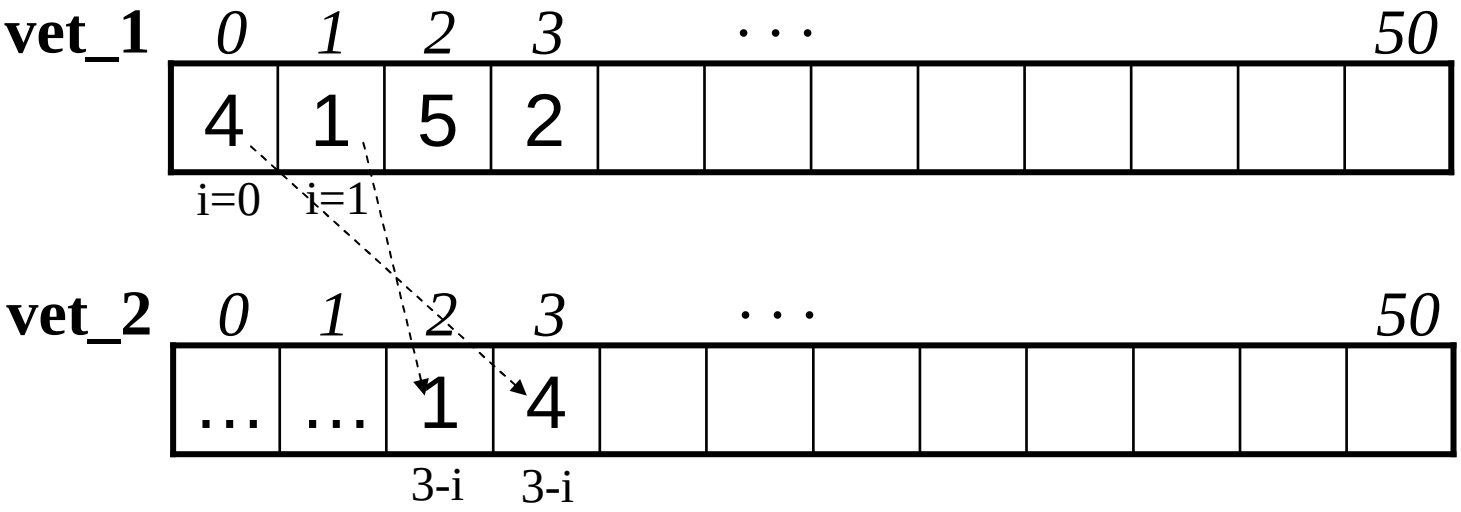
Esercizio 1

Sviluppare un programma che acquisisce dall'utente al massimo 50 numeri interi positivi (interrompendo l'acquisizione se viene inserito il numero 0), li inserisce in un vettore *vet_1* e produce un vettore *vet_2* che li contiene in ordine inverso.

Esercizio 1

Sviluppare un programma che acquisisce dall'utente al massimo 50 numeri interi positivi (interrompendo l'acquisizione se viene inserito il numero 0), li inserisce in un vettore *vet_1* e produce un vettore *vet_2* che li contiene in ordine inverso.

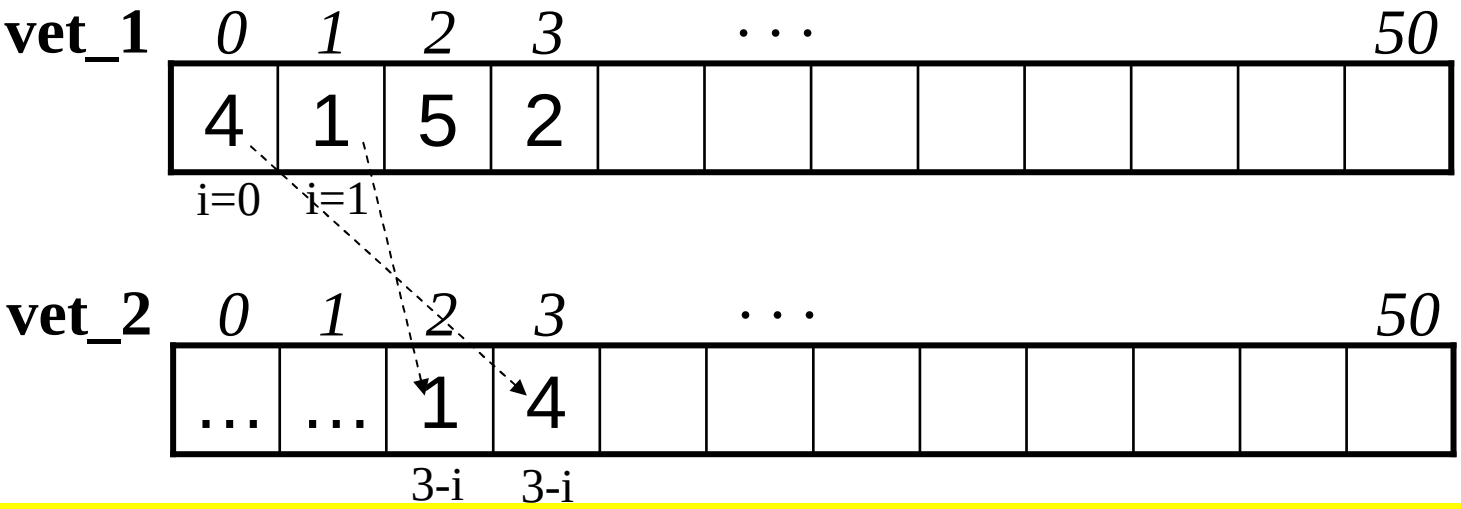
Algoritmo (per la parte di inversione)



Esercizio 1

Sviluppare un programma che acquisisce dall'utente al massimo 50 numeri interi positivi (interrompendo l'acquisizione se viene inserito il numero 0), li inserisce in un vettore *vet_1* e produce un vettore *vet_2* che li contiene in ordine inverso.

Algoritmo (per la parte di inversione)



Supponiamo *n* sia l'ultima posizione del vettore (3 in questo caso):

```
for(i=0; i<=n; i++)
        vet_2[n-i]=vet_1[i];
```

```
printf("Inserisci al massimo 50 numeri positivi (0 per terminare)\n");
i=0;     // prossima posizione libera da occupare
do{
    scanf("%d", &num);
    if(num!=0){
       vet_1[i]=num;
       i++;
       }
} while(num!=0 && i<50);
n=i-1; //ora n indica l'ultima posizione occupata del vettore

for(i=0; i<=n; i++)
   vet_2[n-i]=vet_1[i];

printf("Vettore invertito\n");
for(i=0; i<=n; i++)
  printf("%d\n", vet_2[i]);
```

NB: per forzare l'input di numeri positivi:

```
printf("Inserisci al massimo 50 numeri positivi (0 per terminare)\n");
i=0;     // prossima posizione libera da occupare
do{
   do
      scanf("%d", &num);
   while(num<0)
   if(num!=0){
      vet_1[i]=num;
      i++;
      }
} while(num!=0 && i<50);
n=i-1; //ora n indica l'ultima posizione occupata del vettore
…
```

Esercizio 2

Sviluppare un programma che acquisisca da tastiera due array contenenti 10 numeri interi (int num1[10], int num2[10]), assicurandosi (per ogni array) che l'utente non inserisca un numero già inserito (in questo caso, ripetere l'acquisizione di ogni elemento che sia già stato inserito).

Si trovino quindi tutti gli elementi comuni ad entrambi gli array e si stampi un messaggio indicante, per ogni elemento comune, l'indice occupato nel primo array e nel secondo.

Acquisizione di un vettore con esclusione di numeri già inseriti

```
for(i=0; i<10; i++){
    do
        scanf("%d", &num);
    while(<numero ripetuto>);
    v1[i]=num;
}
```

Acquisisci in num un numero che non sia già stato inserito

Acquisizione di un vettore con esclusione di numeri già inseriti

```
for(i=0; i<10; i++){
    do
        scanf("%d", &num);
    while(<numero ripetuto>);
    v1[i]=num;
}
```

Acquisisci in num un numero che non sia già stato inserito

Raffiniamo il codice in modo da calcolare nella variabile *ripetuto* la condizione da verificare

```
for(i=0; i<10; i++){
    do{
        scanf("%d", &num);
        ripetuto=0;
        for(j=0; j<i; j++)
            if(v1[j]==num) ripetuto=1;
    }while(ripetuto);
    v1[i]=num;
}
```

Identificazione degli elementi in comune tra v1 e v2

Per ogni elemento v1[i] con i=0, …9

- scorri tutti gli elementi v2[j] con j=0..9

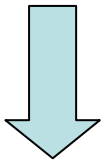
  - se v1[i]= v2[j] allora stampa messaggio

Identificazione degli elementi in comune tra v1 e v2

Per ogni elemento v1[i] con i=0, …9

- scorri tutti gli elementi v2[j] con j=0..9

- se v1[i]= v2[j] allora stampa messaggio



```
for(i=0;i<10;i++)
    for(j=0; j<10; j++)
      if(v1[i]==v2[j])
         printf("Elemento in comune %d, in posizione %d e %d\n", v1[i], i, j);
```

IL CODICE COMPLETO…

```
#include <stdio.h>

main(){
  int v1[10], v2[10];
  int i, j, comune;
  int num;

  printf("Inserisci il primo vettore (10 numeri interi)\n");
  for(i=0;i<10;i++){
     do{
       scanf("%d",&num);
       comune=0;
       for(j=0; j<i; j++)
          if(v1[j]==num)
             comune=1;
     }while(comune);
     printf("Inserito numero %d\n", num);
     v1[i]=num;
     }

printf("Inserisci il secondo vettore (10 numeri interi)\n");
  for(i=0;i<10;i++){
     do{
       scanf("%d",&num);
       comune=0;
       for(j=0; j<i; j++)
          if(v2[j]==num)
             comune=1;
     }while(comune);
     printf("Inserito numero %d\n", num);
     v2[i]=num;
     }

  for(i=0;i<10;i++)
    for(j=0; j<10; j++)
      if(v1[i]==v2[j])
         printf("Elemento in comune %d, in posizione %d e %d\n", v1[i], i, j);

  system("PAUSE");
  return 0;
}
```

Esercizio 3 (ing. informatica 8 feb 2010)

Si sviluppi un programma in linguaggio C che riceva in ingresso due vettori di interi, ciascuno di 10 elementi. Supponendo che i due vettori siano inseriti già ordinati in modo crescente, il programma deve creare e stampare un terzo vettore che rappresenti la "fusione" dei due vettori acquisiti, ovvero che contenga tutti i 20 elementi ordinati tra loro in modo crescente.

Ad esempio, se il primo vettore contiene gli elementi

2 5 9 14 15 20 25 27 30 32

e il secondo vettore contiene gli elementi

3 5 10 11 12 22 23 24 26 27

Il programma crea e stampa un vettore contenente i seguenti elementi:

2 3 5 5 9 10 11 12 14 15 20 22 23 24 25 26 27 27 30 32

Idea chiave

- Ciclo per mettere ordinatamente gli elementi del primo e secondo vettore in un terzo vettore `vett3`
  - mantengo due variabili i e j, che rappresentano l'indice del valore da considerare nel primo e secondo vettore
  - pongo in vett3 l'elemento minore e incremento l'indice corrispondente
  - esco dal ciclo quando uno dei due vettori è stato completamente inserito in `vett3`
- Uscito dal ciclo proseguo inserendo tutti gli elementi del vettore non completamente inserito in `vett3`

Il codice: inserimento dei vettori

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

main(){

  int vett1[10], vett2[10];
  int vett3[20];
  int i, j, z;

  printf("Inserisci il primo vettore\n");
  for(i=0; i<10; i++)
     scanf("%d", &vett1[i]);

  printf("Inserisci il secondo vettore\n");
  for(i=0; i<10; i++)
     scanf("%d", &vett2[i]);
```

Ordinamento “iniziale”

```
i=0;          //elemento da considerare nel primo vettore
j=0;          //elemento da considerare nel secondo vettore
z=0;          //indice del posto libero nel terzo vettore (NB: = i+j)

while(i<10 && j<10)
  if(vett1[i]<vett2[j]){
     vett3[z]=vett1[i];
     i++;
     z++;
     }
     else{
     vett3[z]=vett2[j];
     j++;
     z++;
     }
```

Inserimento finale e stampa

```
if(i<10)
    for(;i<10;i++)
        vett3[z++]=vett1[i];
    else
    for(;j<10;j++)
      vett3[z++]=vett2[j];

   printf("Stampa vettore ordinato\n");
   for(z=0;z<20;z++)
     printf("%d ", vett3[z]);
   printf("\n");

  system("PAUSE");
  return 0;
}
```

Esercizio 4 (Appello 7 aprile 2009)

Scrivere un programma C che:

- richiede all'utente, *in ordine strettamente crescente*, l'inserimento di una serie di numeri interi (al massimo 19), salvandoli nel vettore `num1`. L'acquisizione termina dopo l'inserimento del diciannovesimo numero, o dopo che l'utente inserisce un numero non ordinato (questo numero *non* deve essere salvato);
- stampa a video il vettore acquisito;
- acquisisce dall'utente un numero intero `N` e lo inserisce nel vettore, nella posizione corretta (il vettore deve rimanere in ordine crescente);
- stampa a video il vettore ottenuto.

Il codice: inserimento del vettore ordinato

```
printf("Inserisci in ordine crescente al massimo 19 numeri\n");

scanf("%d", &num);
vett[0]=num;
n=1;                  // indice in cui inserire il prossimo elemento
nonordinato=0;    // nonordinato posto a 1 se inserimento numero
                      // non ordinato
do{
 scanf("%d", &num);
 if(num>vett[n-1]){
   vett[n]=num;
   n++;
   }
 else nonordinato=1;
} while(nonordinato==0 && n<19);
```

Stampa vettore ordinato e acquisizione numero da inserire

```
n=n-1;   // posizione ultimo elemento inserito

for(i=0; i<=n; i++)
  printf("%d ", vett[i]);

printf("Inserisci numero da inserire nel vettore\n");
scanf("%d", &num);
```

Inserimento numero num e stampa finale

```
i=0;
while(i<=n && num>vett[i])
  i++;
//i: posizione in cui inserire l'elemento

for(j=n; j>=i; j--)
  vett[j+1]=vett[j];

vett[i]=num;


for(i=0; i<=n+1; i++)
  printf("%d ", vett[i]);
system("pause");
}
```