



COGNOME E NOME	MATRICOLA	A
AULA	POSTAZIONE	

*Istruzioni per lo svolgimento della prova*

La prova consta di tre quesiti, i primi due da svolgere al calcolatore il terzo su carta.

La soluzione del quesito 1 in modo sufficiente (cioè il programma deve essere funzionante) è **condizione necessaria** per il superamento della prova.

I file .c relativi alla soluzione del quesito 1 e 2 devono essere consegnati su Studium nella sezione elaborati in un unico file compresso (.zip o .rar) contenente i due file chiamato con il vostro numero di matricola (es. o46000123.zip).

### Quesito 1

Completare il seguente codice implementando le funzioni di cui viene fornito il prototipo e completando il codice ove indicato dai commenti. L'esercizio, per essere sufficiente, deve essere compilato e deve poter eseguire.

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

```
/*
La funzione legge un vettore di numeri interi positivi da tastiera fino all'inserimento del
valore 0 o il raggiungimento del numero massimo di elementi n. Restituisce il numero dei valori
effettivamente letti
*/
```

```
int lettura_dati(int* v, int n) {
    for (int i = 0; i < n; ++i) {
        scanf("%d", &v[i]);
        if (v[i] == 0)
            return i;
    }
    return n;
}
```

```
/*
La funzione dato un vettore di ingresso che contiene interi positivi restituisce i valori
strettamente maggiori della soglia nel vettore w
v: vettore di ingresso
w: vettore di uscita (allocato nel main)
m: numero di elementi dei vettori v, w
soglia: soglia
La funzione restituisce (return) è uguale al il numero di elementi significativi presenti in w
*/
```

```
int estrae_numeri_maggiori_di(const int* v, int* w, int m, int soglia) {
    // Il codice deve essere inserito dallo studente
}
```

```
/* La funzione stampa su console il contenuto del vettore */
void stampa( /* i parametri devono essere inseriti dallo studente */) {
    // Il codice deve essere inserito dallo studente
}
```

```
#define DIMENSIONE 10
```

```
int main(void) {
    int v1[DIMENSIONE]; // vettore allocato staticamente
    int *v2;             // vettore allocato dinamicamente
    int dim;             // dimensione del secondo vettore
}
```



```
int dim2;
int s;

//      Lettura da tastiera del primo vettore utilizzando la funzione lettura_dati

dim = lettura_dati( /* lo studente inserisca i parametri corretti */);

/*
    Lo studente deve allocare dinamicamente il secondo vettore in modo che
    abbia tanti elementi quanti sono i valori letti nella funzione precedente.
*/

// inserire il codice corretto per allocare il secondo vettore usando la
// funzione malloc:

/*
    Lo studente deve chiamare la funzione estrae_numeri_maggiori_di salvando
    i valori nel vettore w precedentemente allocato.
*/
printf("inserire una soglia: ");
scanf("%d", &s);
dim2 = estrae_numeri_maggiori_di( /* inserire i parametri corretti */);

/*
    Lo studente, usando la funzione stampa, stampa il risultato contenuto
    in v2
*/
stampa(/* inserire i parametri*/);

// deallocazione
}
```

## Quesito 2

Dato un archivio contenente le informazioni sugli atleti partecipanti ad una gara podistica (numero di gara, cognome, nome, categoria, tempo in secondi), implementare un programma in ANSI C che esponga le funzioni di seguito elencate. Progettare la struttura dati adatta a mantenere in memoria centrale le informazioni utilizzando strutture e puntatori per le liste.

Scrivere un opportuno **main** in cui sia previsto un menu di scelta delle operazioni richieste. Tutti i valori necessari al funzionamento devono essere passati utilizzando parametri, **non è permesso l'uso di variabili globali**.

1. Caricamento dati da file (vedi di seguito) in una lista semplice ordinata rispetto al cognome.
2. Conteggio degli atleti appartenenti ad una categoria il cui valore è fornito come parametro alla funzione.
3. Calcolo della classifica in base ad una categoria fornita come parametro, il risultato deve essere restituito in una seconda lista ordinata (che contiene, quindi, tutti e solo gli atleti che appartengono ad una specifica categoria) in modo decrescente rispetto al tempo, N.B. il contenuto della lista in ingresso NON deve essere né modificato né cancellato, la lista in uscita contiene una copia di tutti gli elementi
4. Funzione che restituisce in un vettore di interi (passato come parametro) il numero di atleti iscritto in ogni categoria;

### Esempio di file

100	BIANCHI	GIUSEPPE	SENIOR	654
12	ROSSI	CARMELO	JUNIOR	513
14	VERDI	GIOVANNI	JUNIOR	495
120	ARANGIO	AGATINO	SENIOR	712
25	AZZURRO	SALVATORE	JUNIOR	501

## Quesito 3

Indicare almeno due algoritmi di ordinamento specificando eventuali condizioni perché sia applicabile e la complessità media e descrivere l'algoritmo di uno dei due.



COGNOME E NOME	MATRICOLA	B
AULA	POSTAZIONE	

*Istruzioni per lo svolgimento della prova*

La prova consta di tre quesiti, i primi due da svolgere al calcolatore il terzo su carta.

La soluzione del quesito 1 in modo sufficiente (cioè il programma deve essere funzionante) è **condizione necessaria** per il superamento della prova.

I file .c relativi alla soluzione del quesito 1 e 2 devono essere consegnati su Studium nella sezione elaborati in un unico file compresso (.zip o .rar) contenente i due file chiamato con il vostro numero di matricola (es. o46000123.zip).

### Quesito 1

Completare il seguente codice implementando le funzioni di cui viene fornito il prototipo e completando il codice ove indicato dai commenti. L'esercizio, per essere sufficiente, deve essere compilato e deve poter eseguire.

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

```
void LeggiVettore(int* v, int n)
{
    for (int i=0; i<n; i++) {
        printf("Inserisci elemento di indice %d: ", i);
        scanf("%d", &v[i]);
    }
}
```

```
/* Visualizza i valori presenti in v di dimensione n */
void Stampa(int *v, int n){
    // Inserire il codice mancante
}
```

```
/* La funzione ConcatenaPositivi, dati due vettori di interi restituisce un nuovo vettore
contenente i valori positivi contenuti nei due vettori. Il vettore va allocato della dimensione
esatta all'interno della funzione e restituito al main (return) */
int* ConcatenaPositivi(...)
```

```
int main(void) {
    int *vett1, *vett2, *vett3;
    int dim1, dim2, dim3;

    printf("Inserisci le dimensioni del vettore1 e del vettore2: ");
    scanf("%d %d", ...);

    vett1 = ...; // Alloca un vettore di dim1 interi
    vett2 = ...; // Alloca un vettore di dim2 interi

    LeggiVettore(...); // Acquisisce il vettore vett1
    LeggiVettore(...); // Acquisisce il vettore vett2

    vett3 = ConcatenaPositivi(...); // Concatena il vettore vett1 e vett2 in vett3

    Stampa(); // Stampa il vettore vett3

    // Disalloca vett1, vett2, vett3
}
```



## Quesito 2

Un ente di formazione eroga 10 corsi diversi ciascuno dei quali è identificato da un codice numerico compreso fra 0 e 9. In un file di testo sono memorizzati uno per riga gli iscritti ai corsi. Per ogni iscritto nel file (vedi esempio) sono memorizzati:

- cognome (una stringa di al massimo 20 caratteri utili senza bianchi)
- nome (una stringa di al massimo 20 caratteri utili senza bianchi)
- età (un intero)
- codiceCorso

Una stessa persona non può essere iscritta due volte allo stesso corso ma può essere iscritta a corsi diversi.

Progettare la struttura dati adatta a mantenere in memoria centrale le informazioni utilizzando strutture e puntatori per le liste;

Scrivere un opportuno *main* in cui sia previsto un menu di scelta delle operazioni richieste. Tutti i valori necessari al funzionamento devono essere passati utilizzando parametri, non è permesso l'uso di variabili globali;

1. Implementare una funzione che esegua il caricamento in un vettore di liste (una lista per ogni corso) il contenuto del file verificando che non ci siano duplicati sullo stesso corso
2. Implementare una funzione che restituisce in un vettore di float l'età media degli iscritti a ciascun corso
3. Implementare una funzione che dato un cognome e nome restituisca la lista dei corsi a cui questa persona è iscritta
4. Implementare una funzione che stampi tutti gli iscritti ad un corso scelto dall'utente

### Esempio dei dati nel file di ingresso

Giuseppe	Verdi	23	1
Alfio	Pappalardo	22	3

## Quesito 3

Scrivere una funzione che dato un albero binario di ricerca contenente come informazione un valore intero restituisca la somma dei valori pari contenuti nei nodi



COGNOME E NOME	MATRICOLA	C
AULA	POSTAZIONE	

*Istruzioni per lo svolgimento della prova*

La prova consta di tre quesiti, i primi due da svolgere al calcolatore il terzo su carta.

La soluzione del quesito 1 in modo sufficiente (cioè il programma deve essere funzionante) è **condizione necessaria** per il superamento della prova.

I file .c relativi alla soluzione del quesito 1 e 2 devono essere consegnati su Studium nella sezione elaborati in un unico file compresso (.zip o .rar) contenente i due file chiamato con il vostro numero di matricola (es. o46000123.zip).

### Quesito 1

Completare il seguente codice implementando le funzioni di cui viene fornito il prototipo e completando il codice ove indicato dai commenti. L'esercizio, per essere sufficiente, deve essere compilato e deve poter eseguire.

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

```
/* La funzione LeggiVettore che consente la lettura di un vettore di interi di dimensione prefissata */
```

```
void LeggiVettore(int *v, int n){
//inserire il codice mancante
}
void Stampa(int *v, int n){
    for (int i=0; i<n; i++) printf("%d",v[i])
}
```

```
/* La funzione Filtra data un vettore di interi restituisce un nuovo vettore contenente solo i valori positivi e la sua dimensione. Il vettore va allocato della dimensione esatta all'interno della funzione e restituito al main. */
```

```
int* Filtra(...){
.....
}
```

```
int main(void) {
    int *v1          // primo vettore
    int *v2;         //secondo vettore

    int dim1         // dimensione del primo vettore
    int dim2         // dimensione del secondo vettore

    LeggiVettore(...);

    v2 = Filtra(.....) //Crea il vettore v2 eliminando i valori negativi v1

    Stampa(.....)     //stampa il vettore v2
}
```



## Quesito 2

All'imbarco di un traghetto possono incolonnarsi 3 tipi di veicoli ciascuno dei quali occupa uno spazio diverso sul traghetto. Ogni veicolo è identificato dalla targa costituita da una stringa di 10 caratteri utili e dal suo tipo.

Progettare la struttura dati adatta a mantenere in memoria centrale costituita da un vettore di tre code una per ogni tipo di veicolo. Scrivere un opportuno **main** in cui sia previsto un menu di scelta delle operazioni richieste. Tutti i valori necessari al funzionamento devono essere passati utilizzando parametri, non è permesso l'uso di variabili globali;

1. Implementare una funzione che consenta di aggiungere un nuovo veicolo nella coda corrispondente. I dati del veicolo vengono inseriti da tastiera e passati come parametro alla funzione;
2. Implementare una funzione che data una targa verifica se il veicolo è presente in una delle tre code. La funzione deve restituire il tipo di veicolo se il veicolo è in coda, -1 se il veicolo non è in coda
3. Implementare una funzione che restituisce un vettore contenente la lunghezza di ciascuna coda
4. Implementare una funzione che restituisce la lista dei veicoli da imbarcare, 5 per ogni tipo. In caso il numero di veicoli sia inferiore a 5, vengono imbarcati solo quelli presenti. I veicoli che vengono imbarcati sono eliminati dalle rispettive code.

## Quesito 3

Descrivere la ricerca binaria su un vettore di interi. Mostrare quali sono gli elementi visitati quando il valore cercato è 3 e l'array contiene i seguenti valori: 4, 9, 12, 15, 17, 20, 25, 30, 35, 38



COGNOME E NOME	MATRICOLA	D
AULA	POSTAZIONE	

*Istruzioni per lo svolgimento della prova*

La prova consta di tre quesiti, i primi due da svolgere al calcolatore il terzo su carta.

La soluzione del quesito 1 in modo sufficiente (cioè il programma deve essere funzionante) è **condizione necessaria** per il superamento della prova.

I file .c relativi alla soluzione del quesito 1 e 2 devono essere consegnati su Studium nella sezione elaborati in un unico file compresso (.zip o .rar) contenente i due file chiamato con il vostro numero di matricola (es. o46000123.zip).

### Quesito 1

Completare il seguente codice implementando le funzioni di cui viene fornito il prototipo e completando il codice ove indicato dai commenti. L'esercizio, per essere sufficiente, deve essere compilato e deve poter eseguire.

```
#include <stdio.h>
#include <stdlib.h>
void InserisciVettore(int v[], int dim) {
    int i;
    for (i = 0; i < dim; i++) {
        printf("Inserisci l'elemento di indice %d: ", i);
        scanf("%d", &v[i]);
    }
}
/* La funzione CercaDuplicati, dato il vettore v, deve individuare tutti gli elementi duplicati
di v ed inserirli in un vettore di dimensione opportuna allocato all'interno della funzione e
restituito (return). Restituisce in dimvd il numero di elementi contenuti nel suddetto
vettore. */
int* CercaDuplicati(int v[], int dimv, int* dimvd) {
    ...
}
/* Visualizza i dati del vettore fornito in ingresso */
StampaVettore(...);

int main(void) {
    int *vett;           /* primo vettore */
    int *duplicati;      /* vettore dei duplicati */
    int dim;             /* dimensione del primo vettore */
    int nduplicati;      /* dimensione del vettore dei duplicati */
    printf("Inserisci il numero di elementi del vettore: ");
    scanf("%d", &dim);

    /* alloca il vettore */
    ...
    /* inserisce gli elementi del vettore */
    InserisciVettore(...);
    /* chiama la funzione CercaDuplicati */
    ... = CercaDuplicati(...);
    /* visualizza il vettore iniziale ed il vettore ottenuto come risultato */
    StampaVettore(...);
    StampaVettore(...);
    /* disalloca i vettori */
    ...
    return 0;
}
```



## Quesito 2

Una azienda si è dotata di un sistema di gestione degli ingressi/uscite dei propri dipendenti dalla propria sede di lavoro. Il sistema, per ogni ingresso/uscita di un dipendente memorizza giornalmente in un file le seguenti informazioni:

- Numero della porta attraversata
- Ora e minuto del passaggio (rappresentati come due interi)
- Codice dipendente (stringa)
- Ingresso/uscita (1 per ingresso, 0 per uscita)

L'azienda è dotata di 3 porte numerate rispettivamente da 0 a 2.

Scrivere un opportuno **main** in cui sia previsto un menu di scelta delle operazioni richieste. Tutti i valori necessari al funzionamento devono essere passati utilizzando parametri, **non è permesso l'uso di variabili globali**.

Implementare le seguenti funzioni:

1. Lettura dei dati di ingresso/uscita dei dipendenti dal file (vedi di seguito) e loro inserimento in un vettore di 3 liste in cui ciascuna lista è associata ad una porta dell'azienda.
2. Funzione che, dato in ingresso un codice di un dipendente, calcoli e restituisca quanto tempo il dipendente è stato in azienda. La funzione restituisca degli opportuni codici di errore se il codice del dipendente non è presente in nessuna lista o se si trova solamente la registrazione dell'ingresso o della sola uscita.
3. Funzione che, dato in ingresso l'identificativo di una porta calcoli e restituisca rispettivamente il numero di dipendenti che sono entrati ed usciti da quella porta
4. Funzione che estragga e restituisca in una lista i dati dei dipendenti che sono entrati dopo un certo orario fornito come parametro di ingresso

### Esempio di file

```
0      8 30  TR135 1
0      8 35  GY879 1
1      8 45  RK682 1
1      8 47  QS174 1
2      9 02  HG365 1
2      9 15  NK341 1
1     16 35  GY879 0
0     16 47  QS174 0
0     17 03  TR135 0
1     17 10  HG365 0
1     17 35  NK341 0
1     17 44  RK682 0
```

## Quesito 3

Scrivere una funzione che, dato un albero binario i cui nodi contengono valori interi, calcoli e restituisca la somma dei valori positivi contenuti nei nodi.





COGNOME E NOME	MATRICOLA	E
AULA	POSTAZIONE	

*Istruzioni per lo svolgimento della prova*

La prova consta di tre quesiti, i primi due da svolgere al calcolatore il terzo su carta.

La soluzione del quesito 1 in modo sufficiente (cioè il programma deve essere funzionante) è **condizione necessaria** per il superamento della prova.

I file .c relativi alla soluzione del quesito 1 e 2 devono essere consegnati su Studium nella sezione elaborati in un unico file compresso (.zip o .rar) contenente i due file chiamato con il vostro numero di matricola (es. o46000123.zip).

### Quesito 1

Completare il seguente codice implementando le funzioni di cui viene fornito il prototipo e completando il codice ove indicato dai commenti. L'esercizio, per essere sufficiente, deve essere compilato e deve poter eseguire.

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

void LeggiVettore(int v[], int dim){
    for (int i=0; i<dim; i++)    {
        printf("Inserisci l'elemento di indice %d: ", i);
        scanf("%d", &v[i]);
        if (i != 0 && v[i] < v[i-1])    {
            printf("Gli elementi devono essere crescenti\n");
            i--;
        }
    }
}

/* La funzione Fusione, dati i due vettori v1 e v2 di dimensione d1 e d2, crea e restituisce un
vettore ordinato in maniera crescente contenente gli elementi di v1 e v2 */
int* Fusione(int v1[], int d1, int v2[], int d2) {
    ...
}

// Visualizza gli elementi del vettore passato come parametro
void Visualizza(...);

int main(void) {
    int *vett1, *vett2, *vettf;
    int dim1, dim2;

    printf("Inserisci le dimensioni dei due vettori: ");
    scanf("%d %d", &dim1, &dim2);

    vett1 = ...; // alloca un vettore di dim1 elementi interi;
    vett2 = ...; // alloca un vettore di dim2 elementi interi;
    LeggiVettore(...); // Acquisisce gli elementi di vett1
    LeggiVettore(...); // Acquisisce gli elementi di vett2

    vettf = Fusione(...); // Fonde vett1 e vett2 formando vettf

    Visualizza(...); // Visualizza vettf

    ...; // Disalloca vett1, vett2, e vettf
    return 0;
}
```



## Quesito 2

Una stampante di rete memorizza per ogni stampa effettuata da un utente remoto le seguenti informazioni:

- Identificativo dell'utente che ha inviato la stampa
- Data della stampa
- Orario della stampa
- Numero di pagine stampate
- Tipo di stampa (fronte e retro o solo fronte)

Scrivere un opportuno *main* in cui sia previsto un menu di scelta delle operazioni richieste. Tutti i valori necessari al funzionamento devono essere passati utilizzando parametri, **non è permesso l'uso di variabili globali**.

1. Implementare una funzione che legga da file l'elenco delle stampe effettuate e le inserisca in una lista di stampe ordinata per data e ora di stampa. (Suggerimento: rappresentare data e ora come stringhe rispettivamente nel formato `aaaa/mm/gg` e `hh:mm:ss`).
2. Implementare una funzione che restituisca il numero di stampe fronte e retro ed il numero di stampe solo fronte effettuate da un utente passato come parametro.
3. Implementare una funzione che restituisca in un opportuno vettore le informazioni delle stampe effettuate in una certa data passata come parametro.
4. Implementare una funzione che crei e restituisca in una lista l'elenco delle stampe effettuate da un certo utente passato come parametro.

## Quesito 3

Scrivere una funzione che dato un albero binario di ricerca di elementi interi, restituisca l'elemento massimo.



COGNOME E NOME	MATRICOLA	F
AULA	POSTAZIONE	

*Istruzioni per lo svolgimento della prova*

La prova consta di tre quesiti, i primi due da svolgere al calcolatore il terzo su carta.

La soluzione del quesito 1 in modo sufficiente (cioè il programma deve essere funzionante) è **condizione necessaria** per il superamento della prova.

I file .c relativi alla soluzione del quesito 1 e 2 devono essere consegnati su Studium nella sezione elaborati in un unico file compresso (.zip o .rar) contenente i due file chiamato con il vostro numero di matricola (es. o46000123.zip).

### Quesito 1

Completare il seguente codice implementando le funzioni di cui viene fornito il prototipo e completando il codice ove indicato dai commenti. L'esercizio, per essere sufficiente, deve essere compilato e deve poter eseguire.

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

```
void LeggiVettore(int v[], int dim){
    for (int i=0; i<dim; i++){
        printf("Inserisci l'elemento di indice %d: ", i);
        scanf("%d", &v[i]);
        if (i != 0 && v[i] < v[i-1]) {
            printf("Gli elementi devono essere crescenti\n");
            i--;
        }
    }
}
```

/\* La funzione Inverti, dato un vettore di interi di dimensione d1 crea e restituisce un vettore v2 di dimensione v2 contenente tutti i valori positivi presenti in v1 ma inseriti dall'ultimo al primo \*/

```
void Inverti(int v1[], int d1, int v2[], int d2) {
    ...
}
// Visualizza gli elementi del vettore passato come parametro
void Visualizza(...);
```

```
int main(void) {
    int *vett1, *vett2,
    int dim1, dim2;

    printf("Inserisci la dimensione del vettore vett1: ");
    scanf("%d", &dim1);

    vett1 = ...; // alloca un vettore di dim1 elementi interi;
    LeggiVettore(...); // Acquisisce gli elementi di vett1
    ... /*alloca il vettore v2 */

    Inverti(...); // Filtra elementi da vett1 copiandoli dall'ultimo al primo

    Visualizza(...); // Visualizza vettf

    ...; // Disalloca vett1, vettf
    return 0;
}
```



## Quesito 2

Si sviluppi un programma console in ANSI C per la gestione di una cantina. La Cantina produce 5 etichette di vini e deve gestire correttamente lo stoccaggio del prodotto finito che avviene, dopo il momento dell'imbottigliamento, numerando le bottiglie ed inserendole nei magazzini differenti in base all'etichetta. Le etichette sono identificate da un numero intero compreso tra 0 e 4.

Specifica della struttura dati

- Etichetta (intero)
- Nome del vino (stringa di 15 caratteri utili senza spazi bianchi)
- Codice bottiglia (stringa di 7 caratteri utili senza spazi bianchi)
- Costo

Scrivere un opportuno **main** in cui sia previsto un menu di scelta delle operazioni richieste. Tutti i valori necessari al funzionamento devono essere passati utilizzando parametri, **non è permesso l'uso di variabili globali**.

Elenco delle operazioni/funzioni che devono essere implementate

1. Funzione di caricamento da file di testo delle bottiglie in un vettore di 5 liste in cui i vini sono suddivisi in base all'etichetta
2. Funzione che dato il numero di etichetta stampa tutte le bottiglie di quella etichetta.
3. Funzione che restituisce il numero di etichetta per la quale sono disponibili più bottiglie.
4. Funzione che restituisce in un vettore la media dei costi per ogni etichetta.

### Esempio di file

0	Etna_Rosso	CE001	5.0
1	Catarratto	E012	8.5
2	Insolia	E005	10.0
1	Pytos	E009	12.5

## Quesito 3

Scrivere una funzione che dato un albero binario di ricerca di elementi interi, restituisca una lista ordinata contenente i valori presenti nell'albero. Si supponga di avere a disposizione il tipo lista e la funzione di inserimento in lista ordinata.