



COGNOME E NOME	MATRICOLA	A
AULA	POSTAZIONE	

#### Istruzioni per lo svolgimento della prova

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

La soluzione del quesito 1 in modo sufficiente (cioè il programma deve compilare ed eseguire correttamente) è condizione necessaria per il superamento della prova.

I file sorgenti (.c) relativi alla soluzione dei quesiti 1 e 2 devono essere consegnati su Studium nella sezione elaborati in un unico file compresso (preferibilmente .zip o .rar) contenente entrambi i file. Inserire all'inizio dei file sorgenti il proprio nome e numero di matricola come commento.

### Quesito 1

Questo esercizio prevede l'uso di codice esistente da integrare e completare in modo da risolvere il problema proposto. Lo studente **deve utilizzare il codice proposto senza alcuna modifica** intervenendo solo nelle parti richieste ed integrandolo se necessario.

Perché l'esercizio sia sufficiente è necessario che il codice compili **senza errori**, il programma esegua senza **errori run-time**, e fornisca il risultato previsto.

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

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

```
void LeggiVettore(int* v, int n)
```

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

```
/* Restituisce un vettore di n elementi. L'elemento di posizione i è la somma degli elementi di posizione da i a n-1  
di v. Cioè detto w il vettore da restituire, w[i] = v[i]+v[i+1]+...+v[n-1] */
```

```
int* SommaElementi(int* v, int n) {...}
```

```
// Visualizza gli elementi di un array passato come parametro
```

```
void Visualizza(...) {...}
```

```
int main(void)
```

```
{  
    int *vett1, *vett2;  
    int dim;  
  
    printf("Inserisci la dimensione del vettore: ");  
    scanf("%d", &dim);  
    // Assicurarsi che dim sia un numero >0 e se non lo è ripetere l'inserimento
```

```
vett1 = ...; // Alloca un array di 'dim' elementi interi  
LeggiVettore(...); // Acquisisce gli elementi in vett1
```

```
vett2 = SommaElementi(...);
```

```
Visualizza(...); // Visualizza vett2
```

```
...; // Disalloca vett1  
...; // Disalloca vett2
```

```
return 0;
```

```
}
```



I QUESITI SUCCESSIVI NON VERRANNO CORRETTI SE IL PROGRAMMA DEL QUESITO 1 NON FUNZIONA CORRETTAMENTE COME DESCRITTO ALL'INIZIO DELL'ESERCIZIO.

## Quesito 2

Un hamburgeria propone un menu di N diversi hamburger (N costante definito a scelta dallo studente). Ogni hamburger è caratterizzato da un nome, un costo, e dall'elenco degli ingredienti. Le informazioni di ogni hamburger sono memorizzate in un elemento di un array di N elementi. Gli ingredienti di un hamburger sono organizzati secondo una lista semplice in cui ogni nodo contiene il nome dell'ingrediente ed il suo costo. Il costo dell'hamburger è calcolato sommando il costo di ogni suo ingrediente.

Lo studente implementi un programma ANSI C con un opportuno *main* in cui sia previsto un menu di scelta delle operazioni richieste e che provvede alla visualizzazione dei risultati quando necessario. Tutti i valori necessari al funzionamento devono essere passati utilizzando parametri, **non è permesso l'uso di variabili globali**.

1. Acquisizione da tastiera di N hamburger. Il primo hamburger inserito viene memorizzato nell'elemento 0 dell'array degli hamburger, il secondo hamburger inserito viene memorizzato nell'elemento 1 dell'array degli hamburger, e così via.
2. Funzione che restituisce un vettore che contiene il numero di ingredienti di ogni hamburger.
3. Funzione che data una fascia di prezzo (passata come parametro) restituisca (e non visualizzi) gli indici dell'array degli hamburger corrispondenti agli hamburger il cui costo cade nella suddetta fascia.
4. Funzione che restituisce l'hamburger di costo massimo, quello di costo minimo ed il costo medio di un hamburger.

## Quesito 3

Descrivere il ruolo dei registri in una CPU.

## Quesito 4

Costruire l'albero binario di ricerca assumendo il seguente ordine di inserimento: 8, 9, 1, 3, 2, 4, 15, 12, 14, 21. Visualizzare la sequenza dei nodi visitati se si effettua la visita anticipata (radice, sinistra, destra).



COGNOME E NOME	MATRICOLA	B
AULA	POSTAZIONE	

#### Istruzioni per lo svolgimento della prova

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

La soluzione del quesito 1 in modo sufficiente (cioè il programma deve compilare ed eseguire correttamente) è condizione necessaria per il superamento della prova.

I file sorgenti (.c) relativi alla soluzione dei quesiti 1 e 2 devono essere consegnati su Studium nella sezione elaborati in un unico file compresso (preferibilmente .zip o .rar) contenente entrambi i file. Inserire all'inizio dei file sorgenti il proprio nome e numero di matricola come commento.

#### Quesito 1

Questo esercizio prevede l'uso di codice esistente da integrare e completare in modo da risolvere il problema proposto. Lo studente **deve utilizzare il codice proposto senza alcuna modifica** intervenendo solo nelle parti richieste ed integrandolo se necessario.

Perché l'esercizio sia sufficiente è necessario che il codice compili **senza errori**, il programma esegua senza **errori run-time**, e fornisca il risultato previsto.

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

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

```
void LeggiVettore(int* v, int n)
```

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

```
/* Restituisce un vettore di m elementi. Il vettore conterrà il quadrato degli elementi positivi presenti nel vettore v */
```

```
int* SommaElementi(int* v, int n, int *m) {...}
```

```
// Visualizza gli elementi di un array passato come parametro
```

```
void Visualizza(...) {...}
```

```
int main(void)
```

```
{  
    int *vett1, *vett2;  
    int dim1, dim2;  
  
    printf("Inserisci la dimensione del vettore: ");  
    scanf("%d", &dim1);  
    // Assicurarsi che dim1 sia un numero >0 e se non lo è ripetere l'inserimento
```

```
vett1 = ...; // Alloca un array di 'dim' elementi interi  
LeggiVettore(...); // Acquisisce gli elementi in vett1
```

```
vett2 = SommaElementi(...);
```

```
Visualizza(...); // Visualizza vett2
```

```
...; // Disalloca vett1  
...; // Disalloca vett2
```

```
return 0;
```

```
}
```



I QUESITI SUCCESSIVI NON VERRANNO CORRETTI SE IL PROGRAMMA DEL QUESITO 1 NON FUNZIONA CORRETTAMENTE COME DESCRITTO ALL'INIZIO DELL'ESERCIZIO.

## Quesito 2

Un negozio di bibite ha memorizzato su un file testo la merce presente nel suo deposito. Per ogni articolo sono presenti nel file le seguenti informazioni:

codice\_articolo

tipo\_bevanda (esempio birra, limonata, etc)

gradazione (per le bevande analcoliche e memorizzato 0.0)

numero\_di\_bottiglie

Lo studente implementi un programma ANSI C con 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 delle informazioni presenti nel file in una lista ordinata rispetto al codice dell'articolo
2. Funzione che restituisce un lista che contiene il numero di bottiglie presenti per ciascuna tipo di bevanda. Nella lista principale (cioè quella ordinata) possono essere presenti bibite diverse dello stesso tipo. Quindi, per esempio, due tipi di birra con diverso codice\_articolo genereranno un solo nodo della lista in cui il numero di bottiglie è la somma del numero di bottiglie dei due tipi di birra.
3. Funzione che suddivide le bevande in 5 fasce di gradazione alcolica (quelle con gradazione uguale a 0, quelle tra 0.1 e 5, quelle tra 5.1 e 10, quelle tra 10.1 e 15 e quelle con gradazione maggiore di 15.1) e restituisce un vettore di 5 elementi contenente il numero di bottiglie presenti nella lista per ciascuna fascia.
4. Funzione che restituisce il codice dell'articolo con la minore e maggiore giacenza.

Struttura del file

A12345 limonata 0.0 45

B34566 birra 4.5 56

13456 birra 7.2 56

67888 chinotto 0.0 47

## Quesito 3

Descrivere il concetto di Bus nell'architettura di Von Neumann e il ruolo di ciascuno di esso.

## Quesito 4

Costruire l'albero binario di ricerca assumendo il seguente ordine di inserimento: 8, 9, 1, 3, 2, 4, 15, 12, 14, 21. Visualizzare la sequenza dei nodi visitati se si effettua la visita per livelli.