

## COMPITI DELLE VACANZE DI INFORMATICA

Classe: 3DLSSA

AS 2017-18

Docente: Buglione Francesca

Per gli studenti con giudizio sospeso: studiare bene input e output di un programma C++, operazioni di selezione (if – if else), cicli (for, while, do while). Studiare le funzioni e la costruzione e stampa di array monodimensionali e bidimensionali. Rifare gli esercizi svolti in classe che sono sul registro elettronico.

Per gli esercizi che seguono, svolgere almeno tanti quanti ne sono indicati in ciascuna unità.

### **UNITA' 1: Funzioni**

Per gli studenti con giudizio sospeso → risolvere almeno 8 esercizi a scelta che seguono

Per gli studenti senza giudizio sospeso → risolvere almeno 5 esercizi a scelta che seguono

## PROBLEMI

Risolvi gli esercizi definendo una funzione per ogni singola attività in modo da ridurre il codice del `main()` a un insieme di chiamate inserite in un ciclo che permetta di ripetere l'esecuzione del programma fino a che l'utente digiti un carattere particolare.

- 1 Leggi una sequenza di numeri; al primo zero incontrato, determina quanti numeri sono stati letti e la loro somma.

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
```

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

- 2 Scrivi un programma che, letto in ingresso il numero di righe da stampare, visualizza i seguenti triangoli famosi:
- 3 Progetta un programma che legge un intero positivo e stampi il suo fattoriale. Si ricorda che il fattoriale  $n!$  è definito come:  $n! = n * (n - 1) * (n - 2) * \dots * 3 * 2 * 1$ .
- 4 Scrivi una procedura/metodo `swap()` che prenda in ingresso due interi e li scambi. Scrivi un programma che utilizzi questa funzione.

- 5 Scrivi un programma che implementi il gioco nel quale l'utente deve indovinare un numero segreto con una quantità massima di tentativi. La funzione/metodo `generaNumero()` permette di generare casualmente un numero di 4 cifre, mentre la funzione/metodo `indovinaNumero()`:
  - consente di inserire un numero;
  - confronta il numero da individuare con quello inserito e visualizza il messaggio "troppo grande" "troppo piccolo";
  - se il numero inserito è corretto, la funzione/metodo ritorna `VERO`, altrimenti ritorna `FALSO`.

Il programma principale controlla se i tentativi ammessi sono esauriti, se il numero segreto è stato individuato o meno, e ripropone una nuova partita.

- 6 Scrivi un programma che permette di calcolare l'area di un cerchio o di un quadrato. L'utente inserisce un numero, dichiarando se si tratta del raggio di un cerchio o del lato di un quadrato. Se l'utente inserisce un numero negativo viene visualizzato un errore, altrimenti il sistema calcola l'area in modo appropriato.

- 7 Leggi due numeri  $n$  e  $m$  ed effettua la moltiplicazione tra due numeri interi utilizzando solo somma, sottrazione, moltiplicazione per 2 e divisione per 2.

- 8 Scrivi un programma ricorsivo che, letto un numero, effettua la somma di tutti i numeri fino a  $n$  a partire dalla seguente pseudocodifica:

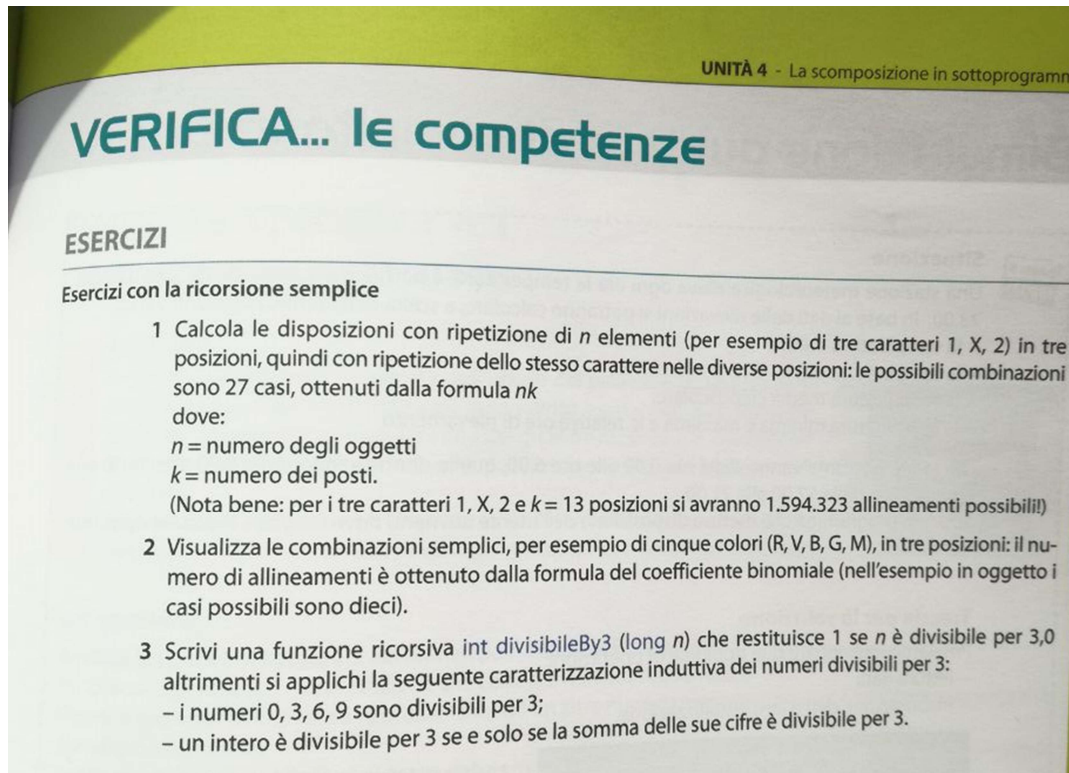
```
int Somma (int N i)
se (N è uguale a 1)
allora
la Somma è il valore del numero
altrimenti
aggiungere all'ultimo elemento la Somma(dei primi N-1 numeri).
```

- 9 Scrivi una funzione ricorsiva che esegue la conversione di un numero da base 10 a base  $n$  leggendo il numero `NUM` e la base  $n$  da input.
- 10 Un intero positivo si dice perfetto se è uguale alla somma dei suoi divisori (sia primi che non primi) escluso se stesso. Per esempio: 28 è perfetto perché  $28 = 1 + 2 + 4 + 7 + 14$ .  
 Scrivi una funzione/metodo `int numeriPerfetti (int n)` che, dato in input un intero positivo  $n$ , restituisca il numero di numeri perfetti tra 1 e  $n$  (estremi inclusi).

## UNITA' 2: Funzioni ricorsive

Per gli studenti con giudizio sospeso → risolvere tutti gli esercizi che seguono

Per gli studenti senza giudizio sospeso → risolvere almeno 1 esercizio a scelta



UNITÀ 4 - La scomposizione in sottoprogrammi

# VERIFICA... le competenze

## ESERCIZI

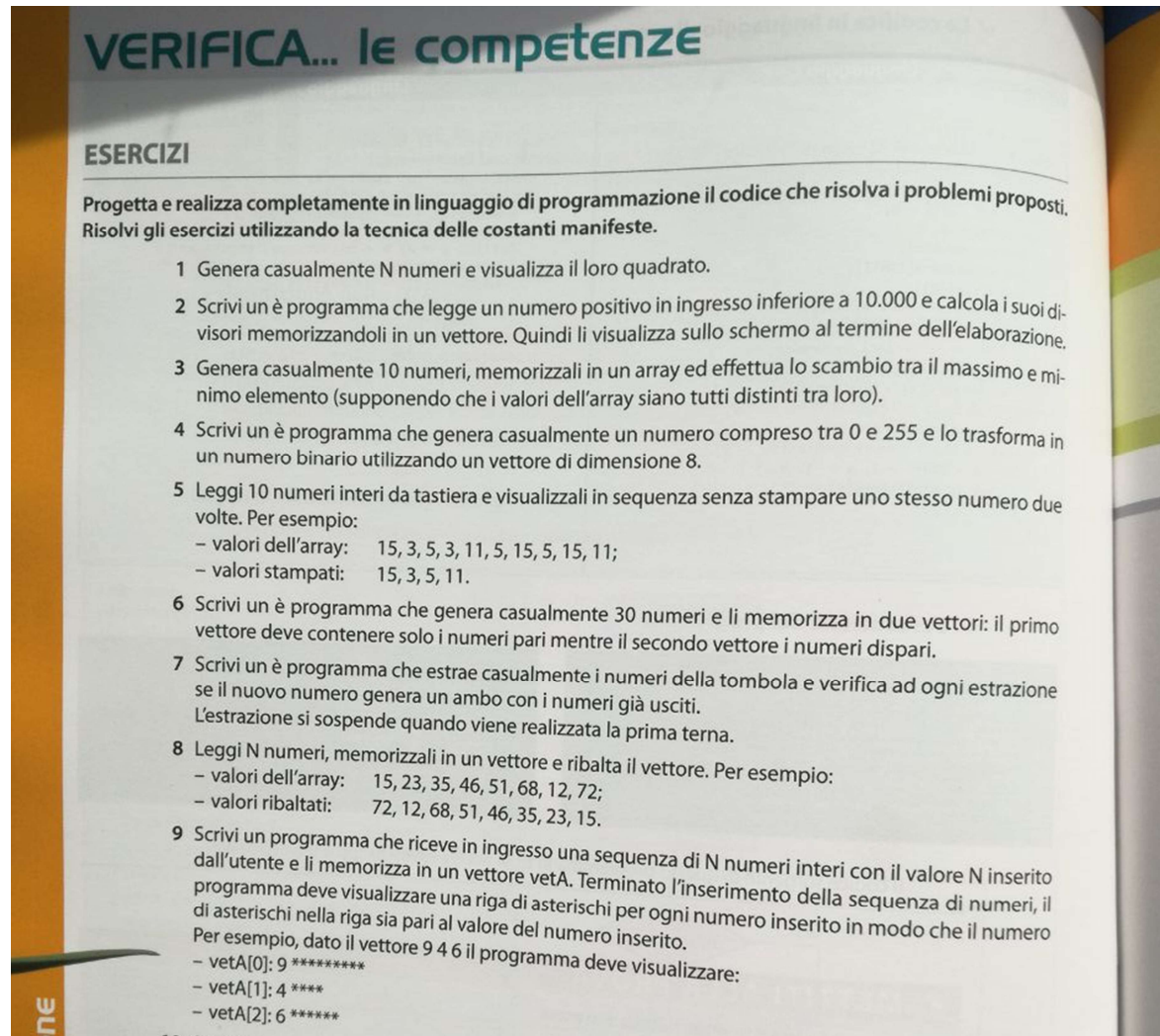
Esercizi con la ricorsione semplice

- 1 Calcola le disposizioni con ripetizione di  $n$  elementi (per esempio di tre caratteri 1, X, 2) in tre posizioni, quindi con ripetizione dello stesso carattere nelle diverse posizioni: le possibili combinazioni sono 27 casi, ottenuti dalla formula  $nk$  dove:  
 $n$  = numero degli oggetti  
 $k$  = numero dei posti.  
(Nota bene: per i tre caratteri 1, X, 2 e  $k = 13$  posizioni si avranno 1.594.323 allineamenti possibili!)
- 2 Visualizza le combinazioni semplici, per esempio di cinque colori (R, V, B, G, M), in tre posizioni: il numero di allineamenti è ottenuto dalla formula del coefficiente binomiale (nell'esempio in oggetto i casi possibili sono dieci).
- 3 Scrivi una funzione ricorsiva `int divisibileBy3 (long n)` che restituisce 1 se  $n$  è divisibile per 3, 0 altrimenti si applichi la seguente caratterizzazione induttiva dei numeri divisibili per 3:
  - i numeri 0, 3, 6, 9 sono divisibili per 3;
  - un intero è divisibile per 3 se e solo se la somma delle sue cifre è divisibile per 3.

### UNITA' 3: Array monodimensionali

Per gli studenti con giudizio sospeso → Svolgere almeno 6 esercizi a scelta fra quelli proposti

Per gli studenti senza giudizio sospeso → Svolgere almeno 4 esercizi a scelta fra quelli proposti



**VERIFICA... le competenze**

**ESERCIZI**

Progetta e realizza completamente in linguaggio di programmazione il codice che risolva i problemi proposti. Risolvi gli esercizi utilizzando la tecnica delle costanti manifeste.

- 1 Genera casualmente N numeri e visualizza il loro quadrato.
- 2 Scrivi un è programma che legge un numero positivo in ingresso inferiore a 10.000 e calcola i suoi divisori memorizzandoli in un vettore. Quindi li visualizza sullo schermo al termine dell'elaborazione.
- 3 Genera casualmente 10 numeri, memorizzali in un array ed effettua lo scambio tra il massimo e minimo elemento (supponendo che i valori dell'array siano tutti distinti tra loro).
- 4 Scrivi un è programma che genera casualmente un numero compreso tra 0 e 255 e lo trasforma in un numero binario utilizzando un vettore di dimensione 8.
- 5 Leggi 10 numeri interi da tastiera e visualizzali in sequenza senza stampare uno stesso numero due volte. Per esempio:  
– valori dell'array: 15, 3, 5, 3, 11, 5, 15, 5, 15, 11;  
– valori stampati: 15, 3, 5, 11.
- 6 Scrivi un è programma che genera casualmente 30 numeri e li memorizza in due vettori: il primo vettore deve contenere solo i numeri pari mentre il secondo vettore i numeri dispari.
- 7 Scrivi un è programma che estrae casualmente i numeri della tombola e verifica ad ogni estrazione se il nuovo numero genera un ambo con i numeri già usciti. L'estrazione si sospende quando viene realizzata la prima terna.
- 8 Leggi N numeri, memorizzali in un vettore e ribalta il vettore. Per esempio:  
– valori dell'array: 15, 23, 35, 46, 51, 68, 12, 72;  
– valori ribaltati: 72, 12, 68, 51, 46, 35, 23, 15.
- 9 Scrivi un programma che riceve in ingresso una sequenza di N numeri interi con il valore N inserito dall'utente e li memorizza in un vettore vetA. Terminato l'inserimento della sequenza di numeri, il programma deve visualizzare una riga di asterischi per ogni numero inserito in modo che il numero di asterischi nella riga sia pari al valore del numero inserito. Per esempio, dato il vettore 9 4 6 il programma deve visualizzare:  
– vetA[0]: 9 \*\*\*\*\*  
– vetA[1]: 4 \*\*\*\*  
– vetA[2]: 6 \*\*\*\*\*



## UNITA' 4: Ordinamento array

Per gli studenti con giudizio sospeso → risolvere tutti gli esercizi che seguono

Per gli studenti senza giudizio sospeso → risolvere almeno 5 esercizi a scelta

UNITÀ 5 - Dati strutturati, file e algoritmi classici

# VERIFICA... le competenze

**AREA DIGITALE**  
Esercizi per l'approfondimento

## ESERCIZI

Progetta il flow-chart e realizza in linguaggio di programmazione il codice che risolve i problemi proposti definendo gli opportuni sottoprogrammi.

- 1 Scrivi un programma che generi casualmente 30 numeri di valore minore di 1000, li memorizzi in un vettore e visualizzi dal più piccolo al più grande i numeri pari.
- 2 Scrivi un programma che generi casualmente TANTI numeri di valore minore di 100 (dove TANTI è una costante manifesta), li memorizzi in un vettore e cancelli i numeri doppi presenti sostituendoli con 0. Quindi li ordini in senso decrescente.
- 3 Scrivi un programma che generi casualmente TANTI numeri di valore minore di 100 (dove TANTI è una costante manifesta), li memorizzi in un vettore e cancelli i numeri doppi presenti sostituendoli con 0. Quindi li ordini in senso crescente ignorando il numero 0 e li visualizzi sullo schermo.
- 4 Scrivi un programma che generi casualmente TANTI numeri di valore minore di 1000 (dove TANTI è una costante manifesta), li memorizzi in un vettore e cancelli i numeri inferiori di un numero MAX letto da tastiera. Quindi visualizzi il vettore ordinato in senso crescente.
- 5 Riordina gli elementi di un vettore di TANTI numeri interi generati casualmente con range  $(-20, +20)$  in modo che tutti i valori negativi siano copiati in un nuovo vettore NEGATIVI e tutti i valori non negativi vengano messi in un vettore POSITIVI. Quindi visualizza i due gruppi di dati ordinati in senso crescente.
- 6 Effettua l'algoritmo di insert-sort su un vettore disordinato dopo aver inserito manualmente i dati senza utilizzare un vettore d'appoggio (ordinamento in sito).
- 7 Riempi in modalità random un vettore di 20 elementi con numeri compresi tra 1 e 100, quindi effettua l'ordinamento mediante l'algoritmo di insert-sort utilizzando un vettore d'appoggio (ordinamento non in sito).
- 8 Effettua l'algoritmo di sele-sort su un vettore disordinato dopo aver inserito manualmente i dati senza utilizzare un vettore d'appoggio (ordinamento in sito).
- 9 Effettua l'ordinamento di un vettore mediante l'algoritmo di merge-sort.

## UNITA' 5: Array bidimensionali

Per gli studenti con giudizio sospeso → risolvere almeno 6 esercizi a scelta

Per gli studenti senza giudizio sospeso → risolvere almeno 3 esercizi a scelta

**VERIFICA... le competenze**

**ESERCIZI**

Progetta e realizza completamente in linguaggio di programmazione il codice che risolva i problemi proposti. Solvi gli esercizi utilizzando, dove possibile, dei sottoprogrammi.

- 1 L'utente inserisce una matrice  $N \times M$  composta da numeri interi. Il programma scambia le righe pari con quelle dispari. Le dimensioni della matrice vengono scelte dall'utente tra i valori massimi fissati all'interno del programma ( $10 \times 20$ ).
- 2 Scrivi un programma che legga da tastiera due matrici  $A$  e  $B$  di  $N \times N$  e calcoli la somma  $C = A + B$  e il prodotto  $D = A \times B$ , visualizzando le matrici ottenute.
- 3 Scrivi un programma che legga una matrice  $A$  ( $M \times P$ ) e una matrice  $B$  ( $P \times N$ ) e calcoli la matrice  $C$  prodotto delle due matrici. La matrice  $C$  è di dimensione  $M \times N$ .
- 4 Data una matrice di  $N \times M$  interi ( $N$  righe,  $M$  colonne):  
`int MAT[N,M] = {3,5,7,9,0,7,2,2,8,9,3,2,8,2,2,2,4,5};`  
scrivi un programma che acquisisca da input due interi, RIGA e COLONNA, e calcoli la somma di tutti gli elementi della sottomatrice ottenuta da MAT eliminando la riga e la colonna indicate da RIGA e COLONNA rispettivamente (cioè la somma di tutti gli elementi di MAT a eccezione di quelli appartenenti alla riga RIGA oppure alla colonna COLONNA).
- 5 Supponi di avere in memoria una matrice rettangolare  $M \times N$  di numeri interi indicanti l'altezza di alcune persone. Qual è la persona più alta: la più bassa tra le più alte di ogni riga, oppure la più alta tra le più basse di ogni colonna?
- 6 Realizza un programma che, partendo da un array che contiene i dati personali dei calciatori di una squadra di calcio, permetta di calcolare:
  - il totale dei goal segnati dalla squadra, specificando:
    - i goal segnati dagli attaccanti (numero di maglia da 7 a 11);
    - i goal segnati dai centrocampisti (numero di maglia da 4 a 6);
    - i goal segnati dai difensori (numero di maglia da 1 a 3);
  - il totale dei goal subiti dalla squadra.Il programma inoltre deve visualizzare i dati personali di tutti i giocatori e per ognuno di essi memorizzare:
  - il numero di maglia;
  - il numero di goal segnati (subiti, se è un portiere).
- 7 Dato il quadrato magico seguente, in cui la somma di righe, colonne e diagonali è uguale a 15, trovane uno diverso:  
4 9 2  
3 5 7  
8 1 6
- 8 Leggi i tempi di 30 concorrenti per due manche di una gara di sci. Dopo la prima manche i concorrenti scendono in ordine inverso rispetto ai tempi ottenuti: leggi i tempi della seconda manche e calcola il tempo totale, quindi visualizza il podio finale (posizione, nome e tempo totale in formato MM.SS.).
- 9 Data una matrice quadrata di ordine  $N$ , verifica se la matrice è diagonale, ricordando che una matrice si dice diagonale se  $A[i, j] = 0$  quando  $i \neq j$ .
- 10 Leggi un numero  $N$  come ordine di una matrice quadrata: successivamente, inserisci i dati interi con il metodo riga  $\times$  colonna. Il programma verifica se tale matrice è simmetrica rispetto alla diagonale principale e successivamente la visualizza.

246



## UNITA' 6: Riepilogo

Per gli studenti con giudizio sospeso → risolvere tutti gli esercizi

Per gli studenti senza giudizio sospeso → risolvere almeno 4 esercizi a scelta

### PROBLEMI

Progetta e realizza completamente in linguaggio di programmazione il codice che risolva i problemi proposti.

- 1 Scrivi un programma che, utilizzando le funzioni, calcola la media e la varianza di un array di numeri reali mediante le seguenti funzioni:
  - `leggi()`: permette di inserire il vettore di numeri;
  - `media()`: calcola e ritorna la media;
  - `varianza()`: calcola e ritorna la varianza;
  - `scrivi()`: visualizza il vettore, la media e la varianza.
- 2 L'utente inserisce due array composti da numeri interi che possono essere di lunghezze differenti. Dopo aver ordinato i due vettori, il programma deve creare un terzo array ordinato a partire dai due array inseriti dall'utente e visualizzarlo sullo schermo.
- 3 In un array di 15 elementi sono memorizzati, ordinati, i 15 migliori tempi ottenuti dagli atleti in una gara di discesa libera. Si riceve poi come dato d'ingresso il tempo ottenuto da un nuovo concorrente: verifica se rientra tra i 15 migliori risultati e, in caso affermativo, inseriscilo nella corretta posizione all'interno del vettore.
- 4 Modifica il programma precedente adattandolo a una gara di slalom dove vengono effettuate due manche: la seconda manche si svolge con l'ordine di partenza stabilito dal peggior tempo della prima manche fino al migliore. Somma i tempi dei concorrenti nelle due manche e visualizza al termine della gara il podio con i tre migliori tempi.
- 5 Un mazzo di 52 carte da poker si può rappresentare mediante 2 array paralleli, il primo per il seme e il secondo per il valore: il valore della i-esima carta del mazzo può assumere i valori da 1 a 13 per, rispettivamente: Asso, 2, ..., 10, J, Q, K, mentre il suo seme può assumere i valori da 1 a 4 per, rispettivamente, Cuori, Quadri, Picche, Fiori. Scrivi una funzione `shuffle()` che mescola e distribuisce il mazzo.
- 6 Scrivi un programma che realizza il seguente gioco tra N giocatori umani e il computer (il banco). Inizialmente ogni giocatore ha un uguale numero di gettoni a ogni mano:
  - ogni giocatore effettua una puntata di x gettoni;
  - il banco distribuisce una carta (la prima del mazzo mescolato) a ognuno degli N giocatori;
  - il giocatore con la carta più alta vince tutte le puntate nella mano corrente;
  - se c'è più di un singolo giocatore con la carta più alta, il banco prende tutto.Il gioco termina quando le carte restanti del mazzo sono meno dei giocatori. Il programma deve simulare il gioco, visualizzando il numero di gettoni a disposizione di ogni giocatore a ogni mano.
- 7 Scrivi un programma per gestire un vettore di record che descrive un insieme di lattine cilindriche atte a contenere alimenti. In particolare, ogni record sarà composto da una stringa atta a contenere una descrizione del contenuto (p.es. "fagioli"), l'altezza della lattina (p.es. 12) e il raggio della lattina (p.es. 7) entrambi in centimetri. Dopo aver riempito il vettore, il programma visualizza i dati in ordine alfabetico oppure solo quello delle lattine con altezza maggiore di 5.

Per gli studenti con giudizio sospeso per la parte di HTML → studiare bene dal sito [w3schools.com](https://www.w3schools.com/html/) le sezioni: HTML Introduction, Editors, Basic, Elements, Attributes, Headings, Paragraphs, Styles, Formatting, Quotations, Comments, Colors, CSS, Links, Images, Tables, Lists, Blocks, Classes, Id.

Svolgere i tutorial proposti e rispondere ai quiz sul sito.