

**A1: PROGRAMMA SVOLTO, ARGOMENTI DI MAGGIOR RILIEVO E COMPITI PER LE VACANZE**

**CLASSE: 3LSA**

**MATERIA: INFORMATICA**

**DOCENTE: SINOPOLI F.**

**1) PROGRAMMA SVOLTO NELL'ANNO SCOLASTICO 2022/2023**

- **C++: LA SCOMPOSIZIONE IN SOTTOPROGRAMMI (Unità 1)**
  - **Funzioni e procedure (L1)**
    - Sottoprogrammi e funzioni
    - Funzioni in linguaggio C++
  - **La condivisione delle variabili nei sottoprogrammi**
    - Modello cliente-servitore
    - Ambiente locale e globale
  - **Le modalità di passaggio dei parametri alle funzioni**
    - Passaggio per valore e per indirizzo
    - Il tipo puntatore
    - Passaggio dei parametri in C++
- **C++: DATI STRUTTURATI E ALGORITMI CLASSICI (Unità 2)**
  - **Dati strutturati semplici: gli array (L1)**
    - Introduzione ai dati strutturati
    - Il vettore o array monodimensionale
    - I vettori in C++
    - Utilizzare i vettori
  - **La ricerca e la disposizione degli elementi in un vettore (L3)**
    - Ricerca sequenziale o lineare
    - Analisi dei dati in un vettore
    - Disposizione dei dati in un vettore
  - **Ordinamento degli elementi in un vettore (L4)**
    - Ordinamento per scambio (bubble sort)
    - Ordinamento per inserimento (insertion sort)
    - Ordinamento per selezione (selection sort)
  - **Due algoritmi evoluti: la ricerca dicotomica e l'ordinamento quicksort (L5)**
    - L'algoritmi di ricerca dicotomica (o binaria)
    - L'algoritmo di ordinamento quicksort
  - **Le matrici, array a 2 dimensioni (L6)**
    - Array a 2 dimensioni
    - Utilizzare le variabili di tipo matrice
    - Definizione di una matrice
    - Utilizzare le matrici rettangolari
    - Matrice quadrata
  - **I record e le tabelle (L7)**
    - Introduzione ai dati strutturati
    - I record e le tabelle in C++

## 2) ARGOMENTI DEL PROGRAMMA DI MAGGIOR RILIEVO:

A PRESCINDERE DAL RIPASSO GENERALE DI TUTTO IL PROGRAMMA SVOLTO SI INDICANO I PUNTI DI MAGGIOR RILIEVO CHE OGNI STUDENTE DEVE RIPASSARE.

- **C++: LA SCOMPOSIZIONE IN SOTTOPROGRAMMI (Unità 1)**
  - **Funzioni e procedure (L1)**
    - Sottoprogrammi e funzioni
    - Funzioni in linguaggio C++
  - **La condivisione delle variabili nei sottoprogrammi**
    - Modello cliente-servitore
    - Ambiente locale e globale
  - **Le modalità di passaggio dei parametri alle funzioni**
    - Passaggio per valore e per indirizzo
    - Il tipo puntatore
    - Passaggio dei parametri in C++
- **C++: DATI STRUTTURATI E ALGORITMI CLASSICI (Unità 2)**
  - **Dati strutturati semplici: gli array (L1)**
    - Introduzione ai dati strutturati
    - Il vettore o array monodimensionale
    - I vettori in C++
    - Utilizzare i vettori
  - **La ricerca e la disposizione degli elementi in un vettore (L3)**
    - Ricerca sequenziale o lineare
    - Analisi dei dati in un vettore
    - Disposizione dei dati in un vettore
  - **Ordinamento degli elementi in un vettore (L4)**
    - Ordinamento per scambio (bubble sort)
    - Ordinamento per inserimento (insertion sort)
    - Ordinamento per selezione (selection sort)
  - **Due algoritmi evoluti: la ricerca dicotomica e l'ordinamento quicksort (L5)**
    - L'algoritmi di ricerca dicotomica (o binaria)
    - L'algoritmo di ordinamento quicksort
  - **Le matrici, array a 2 dimensioni (L6)**
    - Array a 2 dimensioni
    - Utilizzare le variabili di tipo matrice
    - Definizione di una matrice
    - Utilizzare le matrici rettangolari
    - Matrice quadrata
  - **I record e le tabelle (L7)**
    - Introduzione ai dati strutturati
    - I record e le tabelle in C++

**AGLI STUDENTI CHE HANNO LA SOSPENSIONE DEL GIUDIZIO E' RICHiesto UNO STUDIO APPROFONDITO DEGLI ARGOMENTI INDICATI, AL FINE DI COLMARE LE LACUNE MANIFESTATE AL TERMINE DELL'ANNO.**

CLASSE: 3LSA

MATERIA: INFORMATICA

DOCENTE: SINOPOLI F.

**1) COMPITI PER LE VACANZE ESTIVE (PER TUTTI GLI STUDENTI DELLA CLASSE)****Tutti i compiti sono presenti e vanno consegnati su Google Classroom**

Per ciascuno dei seguenti esercizi, sviluppare un programma in C++:

**SELEZIONE / OPERATORI LOGICI / CICLI**

1. Scrivi un programma che, dato in ingresso due numeri interi, determini se:
  - Il primo è multiplo del secondo
  - Il secondo è multiplo del primo
  - Uno è multiplo dell'altro
2. Dato un numero *num*, calcola il numero successivo al doppio del suo quadrato.
3. Dato in ingresso un numero *num*, stampa i primi 10 numeri pari successivi a *num*
4. Leggi una sequenza di numeri crescenti e fermati appena il numero inserito non è maggiore del precedente. Ad esempio: 1,3,5,7,13,17,15.
5. Dati due numeri interi, calcola il minimo comune multiplo

**FUNZIONI**

6. Scrivi un programma in C++ che utilizza funzioni per eseguire le seguenti operazioni:
  - Definisci una funzione *calcolaMedia* che prende in input 3 numeri interi e restituisce la media.
  - Nel *main*, chiedi all'utente di inserire in input i 3 numeri interi.
  - Utilizza la funzione *calcolaMedia* per calcolare la media.
  - Visualizza la media.
7. Scrivi un programma in C++ che utilizza funzioni per eseguire le seguenti operazioni:
  - Definisci una funzione *somma\_cubica* che prende in input 2 numeri interi e restituisce la somma dei loro cubi.
  - Nel *main*, chiedi all'utente di inserire in input i 2 numeri interi.
  - Utilizza la funzione *somma\_cubica* per calcolare la somma dei cubi dei due numeri inseriti.
  - Visualizza il risultato.

8. Scrivi una funzione in C++ chiamata *calcola\_area Rettangolo* che prenda in input la lunghezza e la larghezza di un rettangolo e restituisca l'area corrispondente.
- Successivamente, scrivi un programma principale che soddisfi i seguenti requisiti:
  - Chiedi all'utente di inserire la lunghezza e la larghezza di un rettangolo.
  - Utilizza la funzione *calcola\_area Rettangolo* per determinare l'area del rettangolo.
  - Stampa l'area ottenuta.

## VETTORI

9. 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
10. Scrivi un programma che riempia casualmente un vettore di TANTI numeri garantendo che in esso non siano presenti numeri duplicati
11. 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.
12. Scrivi un programma in C++ che utilizza funzioni per eseguire le seguenti operazioni:
- Definisci una funzione *somma\_cubica* che prende in input 2 numeri interi e restituisce la somma dei loro cubi.
  - Nel *main*, chiedi all'utente di inserire in input i 2 numeri interi.
  - Utilizza la funzione *somma\_cubica* per calcolare la somma dei cubi dei due numeri inseriti.
  - Visualizza il risultato.
13. Scrivi un programma in C++ che soddisfi i seguenti requisiti:
- Dichiarare un array monodimensionale di dimensione 7 chiamato **voti**.
  - Chiedi all'utente di inserire 7 voti interi compresi tra 0 e 10 e memorizzali nell'array **voti**.
  - Calcola e stampa la media dei voti inseriti.
  - Determina e stampa il numero di voti superiori alla media calcolata.
  - Calcola e stampa la percentuale di voti superiori alla media rispetto al totale dei voti inseriti.

## MATRICI

14. Scrivi un programma in C++ che esegua le seguenti operazioni su una matrice quadrata di dimensione 5x5:
- Riempi la matrice con numeri casuali da 1 a 50.
  - Visualizza la matrice.
  - Calcola e visualizza la somma degli elementi presenti sulla diagonale principale.
  - Calcola e visualizza la somma degli elementi presenti sulla diagonale secondaria.
  - Calcola e visualizza la media degli elementi presenti nella prima riga.
15. Scrivi un programma in C++ che esegua le seguenti operazioni su una matrice quadrata di dimensione 3x3:
- Riempi la matrice con numeri casuali da 1 a 10.
  - Visualizza la matrice.
  - Calcola e visualizza la somma degli elementi presenti sulla diagonale principale.
  - Calcola e visualizza la somma degli elementi presenti sulla diagonale secondaria.
  - Calcola e visualizza la somma degli elementi presenti nella prima riga.
  - Calcola e visualizza la somma degli elementi presenti nella prima colonna.
  - Determina e visualizza il valore massimo presente nella matrice.
  - Determina e visualizza il valore minimo presente nella matrice.
16. Scrivi un programma in C++ che soddisfi i seguenti requisiti:
- Dichiarare due matrici quadrate di dimensione 3x3 chiamate **matrice1** e **matrice2**.
  - Chiedi all'utente di inserire gli elementi della matrice **matrice1**.
  - Chiedi all'utente di inserire gli elementi della matrice **matrice2**.
  - Calcola e stampa la somma delle due matrici, cioè una nuova matrice ottenuta sommando gli elementi corrispondenti delle due matrici di input.
  - Calcola e stampa il prodotto delle due matrici, cioè una nuova matrice ottenuta moltiplicando le due matrici di input utilizzando le regole di moltiplicazione tra matrici.
  - Calcola e stampa la trasposta della matrice **matrice1**, cioè una nuova matrice ottenuta scambiando le righe con le colonne.

## RECORD E TABELLE

17. Definisci una struct `s_paziente` per memorizzare i dati dei pazienti di un ospedale. Ogni paziente è caratterizzato dai seguenti attributi:

- Nome del paziente
- Età del paziente
- Altezza del paziente (in centimetri)
- Peso del paziente (in chilogrammi)

Devi scrivere un programma in C++ che soddisfi i seguenti requisiti:

- Chiedi all'utente di inserire i dati di cinque pazienti, ovvero il nome, l'età, l'altezza e il peso di ciascun paziente.
- Visualizza i dati dei pazienti in forma tabellare.
- Determina il paziente più giovane e visualizza il suo nome, l'età, l'altezza e il peso.
- Determina il paziente più alto e visualizza il suo nome, l'età, l'altezza e il peso.
- Calcola l'indice di massa corporea (BMI) di ciascun paziente utilizzando la formula:  $BMI = \text{peso} / (\text{altezza}^2)$ . Determina il paziente con il BMI più alto e, a parità di BMI, il più giovane e visualizza il suo nome, l'età, l'altezza, il peso e il BMI corrispondente

18. Definisci una struct `s_calciatore` per memorizzare i dati dei calciatori. Ogni giocatore è caratterizzato dai seguenti attributi:

- Nome del calciatore
- Età del calciatore
- Altezza del calciatore (in centimetri)
- Peso del calciatore (in chilogrammi)
- Ruolo (P=portiere, D=difensore, C=centrocampista, A=attaccante)

Devi scrivere un programma in C++ che soddisfi i seguenti requisiti:

- Chiedi all'utente di inserire i dati di cinque giocatori, ovvero il nome, l'età, l'altezza, il peso e il ruolo di ciascun calciatore.
- Visualizza i dati in forma tabellare.
- Determina il giocatore più giovane e visualizza il suo nome e la sua età.
- Determina il giocatore più alto e visualizza il suo nome e la sua altezza.
- Determina il portiere più alto, visualizzando il suo nome e la sua altezza.
- Calcola l'indice di massa corporea (BMI) di ciascun paziente utilizzando la formula:  $BMI = \text{peso} / (\text{altezza}^2)$ . Determina il giocatore con il BMI più alto e, a parità di BMI, il più giovane e visualizza il suo nome, l'età, l'altezza, il peso e il BMI corrispondente



**2) GLI STUDENTI CON SOSPENSIONE DEL GIUDIZIO SONO TENUTI A SVOLGERE, OLTRE AI COMPITI DI CUI SOPRA, ANCHE I SEGUENTI ESERCIZI.**

**Tutti i compiti sono presenti nella sezione “Compiti delle vacanze estive | Extra” su Google Classroom**

1. Scrivi un programma che, attraverso una funzione *minimo()*, calcoli il minimo tra due numeri interi che vengono inseriti dall'utente
2. Scrivi un programma che, attraverso una funzione *scambia()*, scambia due numeri che sono letti in input tramite la funzione *inserisciNumero()*
3. Memorizzare in un array di dieci posizioni i primi dieci numeri naturali.
4. Leggere e memorizzare in un array di 8 celle, 8 numeri reali, dopo averli memorizzati calcolarne la somma e la media.
5. Dopo aver letto e memorizzato 8 numeri in un vettore, calcolare la somma di quelli negativi e memorizzare zero al loro posto.
6. Scrivi un programma che:
  - Inserisce 100 numeri naturali, generati casualmente tra 1 e 30, in un vettore
  - Visualizza il vettore
  - Moltiplica per 3 e somma 1 i numeri dispari del vettore
  - Divide per 2 i numeri pari
  - Visualizza il vettore dopo la modifica dei valori
7. Scrivi un programma in C++ che svolga il compito richiesto:
  - Genera un vettore *vettore1* di lunghezza LUNG=50 con numeri casuali da 1 a 100 (0,5 punti)
  - Visualizza *vettore1* (0,5 punti)
  - Visualizza *vettore1* al rovescio, ossia dalla posizione finale alla posizione iniziale (0,5 punti)
  - Ordina il *vettore1* (algoritmo di ordinamento a tua scelta) (1,5 punto)
  - Visualizza il *vettore1* in ordine crescente (0,5 punti)
  - Genera un vettore *vettore2* contenente gli elementi del *vettore1* con un valore maggiore o uguale a 50 (1 punto)
  - Visualizza *vettore2* (0,5 punti)
  - Calcola e visualizza la somma e la media degli elementi del *vettore2* (1 punto)
  - Genera un vettore *vettore3* contenente gli elementi in posizione pari (posizione 0, 2, 4, 6, ....) del *vettore1* (1 punto)
  - Visualizza *vettore3* (0,5 punti)