

Repubblica Italiana Provincia Autonoma di Bolzano - Alto Adige		Republik Italien Autonome Provinz Bozen - Südtirol
<b><i>Istituto d'Istruzione Secondaria Superiore per le scienze, le tecnologie e i servizi</i></b>		
<b><i>"GALILEO GALILEI"</i></b>		
<b><i>Oberschulzentrum für Wissenschaften, Technologie und Dienstleistungen</i></b>		
ISTITUTO TECNICO TECNOLOGICO - LICEO SCIENTIFICO-SCIENZE APPLICATE		
ISTITUTO PROFESSIONALE PER L'INDUSTRIA E L'ARTIGIANATO - ISTITUTO PROFESSIONALE ODONTOTECNICO		
Fachoberschule für den Technologischen Bereich - Realgymnasium mit Schwerpunkt angewandte Naturwissenschaften		
Berufsbildende Oberschule für Industrie und Handel - Berufsbildende Oberschule für Zahntechniker		
39100 BOLZANO- via Cadorna 14 Cod. Fisc. 80006520219		39100 Bozen - Cadornastraße 14 St.Nr. 80006520219

## Programma Svolto

### Indirizzo Informatica e Telecomunicazioni

#### Articolazione Informatica

DOCENTE:	<b>MARZOCHELLA ANTIMO</b>
ITP:	<i>ALOISI MARA</i>
MATERIA:	<i>INFORMATICA</i>
CLASSE:	<i>3° K</i>
ORE SETTIMANALI:	<i>7 (di cui 6 in codocenza)</i>
ANNO SCOLASTICO:	<i>2019/2020</i>

LUOGO E DATA

FIRMA



**BOLZANO, 01/06/2020**

## Competenze finali

◆ **Competenza n. 1**

*Analizzare dati e interpretarli sviluppando deduzioni e ragionamenti sugli stessi anche con l'ausilio di rappresentazioni grafiche, usando consapevolmente gli strumenti di calcolo e le potenzialità offerte da applicazioni specifiche di tipo informatico*

◆ **Competenza n. 2**

*Individuare le strategie appropriate per la soluzione di problemi*

◆ **Competenza n. 3**

scegliere dispositivi e strumenti in base alle loro caratteristiche funzionali

## TAVOLA DI PROGRAMMAZIONE

MATERIA: *INFORMATICA*

CLASSE: 3°

<b>Modulo N° I: Introduzione ai problemi, agli algoritmi ed ai programmi</b>			
Attività/metodologie didattiche	Strumenti didattici	Tipologia verifiche	Tempi: Ore
lezione frontale partecipata, problem-solving, didattica laboratoriale.	libro di testo, appunti del docente, materiale multimediale, laboratorio d'informatica, IDE( <i>Geany (per C/C++)</i> , <i>CodeBlocks</i> ), <i>shell</i> di Linux	Scritte/orali, pratiche	42
Obiettivi disciplinari	Contenuti		
<p><b>COMPETENZA 1:</b> <i>Analizzare dati e interpretarli sviluppando deduzioni e ragionamenti sugli stessi anche con l'ausilio di rappresentazioni grafiche, usando consapevolmente gli strumenti di calcolo e le potenzialità offerte da applicazioni specifiche di tipo informatico;</i></p> <p><b>COMPETENZA 2</b> <i>Individuare le strategie appropriate per la soluzione di problemi</i></p> <p><b>ABILITA'</b> Distinguere tra i concetti di istruzione, algoritmo, programma e processo Distinguere tra problemi decidibili e non decidibili (applicazione della tesi di Church) Saper utilizzare i formalismi per la rappresentazioni degli algoritmi (<i>Flow-chart PDL</i>), essere in grado di darne la specifica Saper analizzare semplici problemi e proporre algoritmi risolutivi Saper verificare le soluzioni trovate, utilizzo delle tavole di traccia per le configurazioni istantanee Saper individuare e correggere gli errori a tempo di compilazione e di esecuzione</p>	<p><b>PROGRAMMA GENERALE</b></p> <ul style="list-style-type: none"> <li>➤ Terminologia generale dell'informatica, elaborazione, automazione e dati; algoritmi e relative tecniche di analisi (<i>Flow-chart PDL</i>)</li> <li>➤ Specifica degli algoritmi, tipi di dati ed istruzioni (<i>programmazione imperativa</i>), strutture dati statiche: array lineari e matrici</li> </ul> <p><b>CONOSCENZE:</b></p> <ul style="list-style-type: none"> <li>➤ termini generali della disciplina; proprietà fondamentali degli algoritmi</li> <li>➤ enunciato della tesi di Church e relative implicazioni nel campo informatico</li> <li>➤ i vari tipi di dati e la sintassi delle diverse istruzioni: assegnamento, input/output, controllo</li> <li>➤ Struttura generale di un programma scritto attraverso un text-editor o un IDE in linguaggio di programmazione "C/C++"</li> <li>➤ errori di sintassi, logici ed in fase di esecuzione</li> <li>➤ significato di puntatore o indirizzo</li> <li>➤ tecniche di analisi della complessità spaziale e temporale degli algoritmi</li> </ul> <p><b>LABORATORIO</b> Verranno utilizzati gli IDE <i>Geany (per C/C++)</i>, <i>CodeBlocks</i> e la <i>shell</i> di Linux per lo sviluppo di semplici programmi secondo il <i>paradigma imperativo</i></p>		

<b>Modulo N° II: programmazione strutturata ed algoritmi notevoli</b>			
Attività/metodologie didattiche	Strumenti didattici	Tipologia verifiche	Tempi: Ore
lezione frontale partecipata, problem-solving, didattica laboratoriale.	libro di testo, appunti del docente, materiale multimediale, laboratorio d'informatica, IDE(Geany (per C/C++), CodeBlocks), shell di Linux	Scritte/orali, pratiche	42
Obiettivi disciplinari		Contenuti	
<p><b>COMPETENZA 1:</b>  <i>Analizzare dati e interpretarli sviluppando deduzioni e ragionamenti sugli stessi anche con l'ausilio di rappresentazioni grafiche, usando consapevolmente gli strumenti di calcolo e le potenzialità offerte da applicazioni specifiche di tipo informatico;</i></p> <p><b>COMPETENZA 2</b>  <i>Individuare le strategie appropriate per la soluzione di problemi</i></p> <p><b>ABILITA'</b></p> <p>Saper utilizzare le principali operazioni su liste lineari statiche: array, record</p> <p>Saper applicare gli algoritmi di ricerca su liste di tipo lineari statiche</p> <p>Riuscire ad applicare gli algoritmi fondamentali di ordinamento sulle strutture lineari</p> <p>Saper utilizzare le funzioni (in C/C++) ovvero in genere i sottoprogrammi allo scopo di modulare la struttura del programma.</p> <p>Saper utilizzare il passaggio dei parametri per valore e per indirizzo comprendendone l'utilità</p> <p>Essere in grado di applicare la ricorsione (metodo top-down) quando la natura bottom-up della soluzione è di difficile impostazione algoritmica</p>		<p><b>PROGRAMMA GENERALE</b></p> <ul style="list-style-type: none"> <li>➤ Strutture dati astratte lineari statiche</li> <li>➤ Metodologia top-down e bottom-up</li> <li>➤ funzioni e sotto-algoritmi/sotto-programmi</li> </ul> <p><b>CONOSCENZE:</b></p> <ul style="list-style-type: none"> <li>➤ Strutture dati lineari statiche: array e record</li> <li>➤ Principali algoritmi di ricerca su strutture dati lineari statiche</li> <li>➤ Problema dell'ordinamento e relativi algoritmi per le applicazioni informatiche</li> <li>➤ Definizione del prototype di una funzione (sotto-algoritmo) e del relativo codice ai fini della programmazione strutturata</li> <li>➤ Significato del passaggio di parametri per valore e per indirizzo</li> </ul> <p>➤</p> <p><b>LABORATORIO</b></p> <p>Verranno utilizzati gli IDE Geany (pe C/C++), CodeBlocks e la shell di Linux per lo sviluppo di semplici programmi secondo il <i>paradigma imperativo</i></p>	

\*\*\*\* DA QUI IN POI LA DIDATTICA SI SVOLGE A DISTANZA \*\*\*\*

<b>Modulo N° III Elementi fondamentali di gestione dei file</b>			
Attività/metodologie didattiche	Strumenti didattici	Tipologia verifiche	Tempi: Ore
lezione con uso di video e/o Videoconferenza, problem-solving, didattica laboratoriale.	libro di testo, appunti del docente, materiale multimediale, laboratorio d'informatica, IDE(Geany (per C/C++), CodeBlocks), shell di Linux	Scritte/orali, pratiche	28
Obiettivi disciplinari		Contenuti	
<p><b>COMPETENZA 1:</b>  <i>Analizzare dati e interpretarli sviluppando deduzioni e ragionamenti sugli stessi anche con l'ausilio di rappresentazioni grafiche, usando consapevolmente gli strumenti di calcolo e le potenzialità offerte da applicazioni specifiche di tipo informatico;</i></p> <p><b>COMPETENZA 2</b>  <i>Individuare le strategie appropriate per la soluzione di problemi</i></p> <p><b>ABILITA'</b></p> <p>Essere in grado di organizzare i dati attraverso i file e di confrontarne le possibili organizzazioni (sequenziale, random, con indici(cenni))            Saper utilizzare le funzioni e le procedure di gestione dei file in linguaggio C/C++            Essere in grado di implementare le operazioni di creazione, scansione, inserimento di record in file</p>		<p><b>PROGRAMMA GENERALE</b></p> <ul style="list-style-type: none"> <li>➤ Concetti elementari fondamentali: file, record, campo</li> <li>➤ Organizzazione logica e fisica di un file</li> <li>➤ File di testo e non</li> <li>➤ Tecniche di accesso ad un file: sequenziale, random o diretto,</li> <li>➤ Operazioni fondamentali e notevoli sui file</li> </ul> <p><b>CONOSCENZE:</b></p> <ul style="list-style-type: none"> <li>➤ Conoscere le caratteristiche dei file, le operazioni possibili su di essi e le problematiche relative alla loro gestione (logica e di S.O.)</li> <li>➤ Conoscere le funzioni e procedure per il trattamento dei file in linguaggio C/C++</li> <li>➤ Conoscere le operazioni di creazione, scansione, inserimento di record in file</li> <li>➤ Conoscere le operazioni di modifica di record, cancellazione</li> </ul> <p><b>LABORATORIO</b></p> <p>Verranno utilizzati gli IDE Geany (pe C/C++), CodeBlocks e la shell di Linux per lo sviluppo di semplici programmi secondo il <i>paradigma imperativo</i></p>	

## **MODALITÀ DI VALUTAZIONE (DaD)**

Vedi delibera del Collegio dei Docenti, nella seduta del 22/04/2020 e la relativa griglia di valutazione di riferimento (Vedi Allegati: [Delibera CD 22-Aprile](#) ).

## **BIBLIOGRAFIA E MATERIALI DIDATTICI**

- Informatica Lorenzi Agostino Moriggia Vittorio Rizzi Andrea *Informatica Per Istituti Tecnici Tecnologici A* Edizioni ATLAS vol. A e materiali forniti dal docente.
- I materiali forniti dal docente sono pubblicati su Registro elettronico/Didattica URL: <https://web.spaggiari.eu/home/app/default/login.php>

**Gli Studenti**

**Insegnanti:**

**A.Marzocchella e M. Aloisi**