6.9 INFORMATICA

I ANNO

Unità didattica	OBIETTIVI		
	Conoscenze	**Competenze**	**Abilità**
Introduzione all'informatica	• Concetti generali su Informatica e Sistemi di elaborazione • Hardware • Software • Classificazione dei tipi di computer • Reti e Internet	• Cogliere i motivi della crescente utilizzazione dei calcolatori e le opportunità legate al loro impiego comprendendo il ruolo esecutivo della macchina	• Padronanza di accesso e utilizzo degli strumenti hardware e software sviluppando buone capacità all'uso delle apparecchiature
La rappresentazione delle informazioni	• Il sistema di numerazione binario • I metodi di rappresentazione dei dati all'interno dell'elaboratore	• Comprendere la logica dei sistemi posizionali • Capire come rappresentare i dati di un sistema di elaborazione	• Essere in grado di eseguire le 4 operazioni in sistema di numerazione diversi da quello decimale • Calcolare l'occupazione di un dato in memoria
Struttura dell'elaboratore	• Il modello logico funzionale • La memoria centrale • La CPU • Altre memorie • Periferiche di input • Periferiche di output	• Capire la logica di funzionamento dei sistemi di elaborazione • Capire le modalità di interazione tra i principali dispositivi hardware di un elaboratore	• Riconoscere all'interno di un elaboratore i principali dispositivi hardware
Primi elementi di programmazione	• Problemi e algoritmi • Componenti di un algoritmo: dati e istruzioni • Rappresentazione e verifica di algoritmi • Descrizione dei programmi • Regole di base di un linguaggio di programmazione • Il linguaggio di programmazione Visual Basic • Saper riconoscere il ruolo esecutivo della macchina • Struttura di sequenza	• Saper usare le istruzioni di lettura, scrittura e assegnazione • Individuare le fasi necessarie per passare da un problema alla sua soluzione • Saper riconoscere i dati di input e i dati di output di un problema	• Saper realizzare algoritmi che prevedono istruzioni di lettura, scrittura e assegnazione • Saper descrivere algoritmi tramite i diagrammi di flusso • Saper scrivere programmi che traducono l'algoritmo in linguaggio di programmazione
La selezione	• Le strutture condizionali semplici • Le strutture condizionali complesse	• Rappresentare le strutture condizionali • Formulare strutture condizionali sintatticamente corrette	• Individuare le strutture di controllo più idonee a risolvere un determinato problema • Codificare programmi che includono le strutture condizionali

II ANNO

Unità didattica	OBIETTIVI		
	Conoscenze	**Competenze**	**Abilità**
Selezioni complesse	• Le basi della logica simbolica e del calcolo proposizionale • Condizioni composte • Selezione multipla	• Rappresentare le strutture condizionali • Formulare strutture condizionali sintatticamente corrette • Applicare agli algoritmi i principi della logica proposizionale	• Individuare le strutture di controllo più idonee a risolvere un determinato problema • Adattare le frasi del linguaggio corrente al formalismo delle proposizioni • Codificare programmi che includono le strutture condizionali
Struttura iterativa	• Le strutture iterative con controllo in testa • Le strutture iterative con controllo in coda • Le strutture iterative con numero prefissato di cicli • Cicli per ricerche complesse • Cicli annidati	• Rappresentare le strutture iterative • Scrivere programmi che contengono cicli	• Individuare il tipo di ciclo più adatto a risolvere un determinato problema • Saper trasformare un ciclo precondizionale in uno postcondizionale • Codificare programmi che includono le strutture iterative • Testare algoritmi creando insiemi significativi di casi prova
Programmazione complessa	• La progettazione Top-down • La scomposizione in sottoproblemi • Le variabili locali e globali • Le procedure e le funzioni	• Saper affrontare un problema scomponendolo in sottoproblemi • Conoscere e saper usare le variabili locali e globali • Saper usare le procedure e le funzioni quando necessario	• Scrivere programmi scomposti in sottoprogrammi • Scrivere programmi che utilizzino le funzioni • Scrivere programmi con variabili globali • Scrivere programmi con procedure in cui si usano variabili locali
Le strutture dati semplici	• Differenza tra variabili semplici e variabili strutturate • Rappresentazione e gestione dei dati con vettori	• Saper caricare e visualizzare dati di un vettore • Saper cercare informazioni all'interno di strutture dati • Saper individuare massimo, minimo e media di un vettore • Saper fornire totali parziali elaborando i dati di un vettore	• Organizzare i dati nei vettori • Applicare gli algoritmi di gestione dati nei vettori
Le strutture dati complesse	• Caratteristiche di una matrice • Rappresentazione e gestione dei dati con una matrice	• Inserire ed effettuare operazioni sugli elementi di una matrice • Risolvere problemi che richiedano l'uso di strutture dati complesse	• Scrivere programmi in grado di trovare massimi, minimi e medie di righe e colonne di una matrice • Scrivere programmi che utilizzino i dati memorizzati in una matrice quadrata