

CLASSE: 2LSA

MATERIA: INFORMATICA

DOCENTE: PUPA

PROGRAMMAZIONE DIDATTICA

 ■ ASSE CULTURALE SCIENTIFICO TECNOLOGICO

competenze chiave	competenze base	conoscenze	abilità
<ul style="list-style-type: none"> • Imparare a imparare • Competenza digitale • Comunicazione e nella madrelingua • Comunicazione e nelle lingue straniere 	<ul style="list-style-type: none"> • Utilizzare algoritmi per la risoluzione di problemi • Comprendere il concetto di variabile e cella di memoria • Utilizzare i diagrammi di flusso per rappresentare gli algoritmi • Realizzare i diagrammi di flusso utilizzando Flowgorithm 	<ul style="list-style-type: none"> • Conoscere la simbologia per progettare e realizzare diagrammi di flusso • Comprendere gli elementi caratterizzanti di un algoritmo • Comprendere la relazione tra algoritmo e programma • Acquisire il concetto di linguaggio di progetto e di pseudocodifica 	<ul style="list-style-type: none"> • Educare al pensiero computazionale per risolvere problemi • Descrivere procedure mediante algoritmi • Rappresentare gli algoritmi mediante flow charts • Memorizzare le informazioni nelle variabili
<ul style="list-style-type: none"> • Competenze di base in scienze e tecnologia • Competenza digitale • Imparare a imparare • Comunicazione e nella madrelingua • Comunicazione e nelle lingue straniere 	<ul style="list-style-type: none"> • Sviluppare capacità di problem solving • Utilizzare linguaggi visuali per scrivere programmi • Saper collaudare un programma • Imparare le fasi di design e coding attraverso il linguaggio visuale Scratch 3 	<ul style="list-style-type: none"> • Conoscere ambienti visuali per realizzare programmi • Apprendere l'anatomia di uno script e di una app 	<ul style="list-style-type: none"> • Creare programmi in autonomia • Realizzare le animazioni nei programmi • Acquisire la predisposizione al passaggio verso linguaggi più complessi
<ul style="list-style-type: none"> • Imparare a imparare • Competenza digitale • Comunicazione e nella 	<ul style="list-style-type: none"> • Scrivere un programma in linguaggio di progetto • Editare, testare e collaudare un 	<ul style="list-style-type: none"> • Conoscere i diversi formati di un programma • Descrivere le diverse fasi di 	<ul style="list-style-type: none"> • Disporre l'output sullo schermo • Utilizzare le variabili nei programmi

<ul style="list-style-type: none"> madrelingua • Comunicazione e nelle lingue straniere 	<ul style="list-style-type: none"> programma • Effettuare l'input dei dati • Formattare l'output numerico sullo schermo • Scrivere programmi con istruzioni in sequenza e in blocchi 	<ul style="list-style-type: none"> sviluppo di un programma • Comprendere il concetto di variabile • Conoscere le istruzioni di comunicazione con l'utente • Comprendere l'importanza del commento del codice 	<ul style="list-style-type: none"> • Commentare il codice del programma • Utilizzare variabili intere e reali • Utilizzare l'operatore MOD sui numeri interi
<ul style="list-style-type: none"> • Imparare a Imparare • Competenza digitale • Comunicazione e nella madrelingua • Comunicazione e nelle lingue straniere 	<ul style="list-style-type: none"> • Riconoscere quando utilizzare selezioni annidate • Codificare la selezione semplice e doppia • Effettuare l'annidamento delle istruzioni • Utilizzare le variabili di tipo bool • Utilizzare gli operatori logici 	<ul style="list-style-type: none"> • Comprendere l'importanza del commento all'interno del codice • Conoscere l'istruzione di selezione semplice e doppia • Conoscere le variabili di tipo booleano e gli operatori logici • Comprendere il concetto di annidamento • Conoscere la notazione per i blocchi di istruzioni 	
<ul style="list-style-type: none"> • Imparare a Imparare • Competenza digitale • Comunicazione e nella madrelingua • Comunicazione e nelle lingue straniere 	<ul style="list-style-type: none"> • Saper scegliere il tipo di iterazione adeguato alle diverse situazioni • Generare numeri casuali • Codificare l'iterazione indefinita • Codificare l'iterazione definita 	<ul style="list-style-type: none"> • Acquisire il concetto di iterazione • Conoscere le diverse tipologie di iterazione • Conoscere la differenza tra controllo di testa e di coda • Conoscere la differenza tra iterazione 	<ul style="list-style-type: none"> • Scrivere programmi con selezioni e iterazioni • Progettare programmi con cicli annidati • Utilizzare diverse tipologie di iterazione nello stesso programma
	<ul style="list-style-type: none"> • Codificare programmi con 	definita e indefinita	<ul style="list-style-type: none"> • Utilizzare la trace table per

	cicli annidati		individuare errori nel codice
--	----------------	--	-------------------------------

CONTENUTI DEL PROGRAMMA:

- Problemi e algoritmi
 - Il problem solving
 - Gli algoritmi
 - Istruzioni e variabili
 - Diagrammi di flusso e pseudolinguaggio
- Flowgorithm
 - Presentazione di Flowgorithm
 - Il costrutto di selezione
 - I connettivi logici
 - Il ciclo for
 - Il ciclo while
- Programmare in C++ e Python
 - I linguaggi per la programmazione degli elaboratori
 - Il programma e le variabili
 - L'input e l'output dei dati
 - Metodologia per la risoluzione di un problema
- La selezione
 - L'istruzione di selezione semplice e doppia
 - La selezione annidata e multipla
 - Gli operatori logici nella selezione
- L'iterazione indefinita e definita
 - L'istruzione di iterazione precondizionata
 - L'istruzione di iterazione post-condizionata
 - L'istruzione di iterazione definita

Attività del docente e metodologia	Attività dello studente	Materiali e spazi utilizzati
<ul style="list-style-type: none"> • Lezioni frontali utilizzando supporti multimediali • Assegnare agli alunni i riferimenti sul libro e i compiti settimanali da svolgere. 	<ul style="list-style-type: none"> • Seguire le spiegazioni del docente, prendendo appunti e riorganizzando il materiale in modo logico • Utilizzare il linguaggio 	<ul style="list-style-type: none"> • Libri di testo, e-book, presentazioni PowerPoint • Laboratori e PC in ambiente Windows e Mac • Proiettore
<ul style="list-style-type: none"> • Assicurarsi che gli studenti comprendano le consegne 	proprio della disciplina esponendo con chiarezza i concetti	<ul style="list-style-type: none"> • Software di laboratorio

<ul style="list-style-type: none"> • Stimolare l'approccio al problem solving • Motivare gli studenti allo studio della materia, stimolando la loro curiosità verso gli argomenti affrontati • Insistere sull'uso di un linguaggio corretto • Fornire gli strumenti utili alla risoluzione di problemi • Riprogrammare le eventuali unità di apprendimento di recupero per gli alunni che non hanno conseguito gli obiettivi programmati 	<ul style="list-style-type: none"> • Partecipare attivamente all'attività didattica, ponendo domande, proponendo soluzioni, riflettendo sugli interventi di docente e compagni • Lavorare in gruppo, condividendo informazioni e azioni e accogliendo suggerimenti o ipotesi alternative 	
---	--	--

VALUTAZIONE:

CONOSCENZE	VALUTAZIONE	ABILITA'	VALUTAZIONE	COMPETENZE	VALUTAZIONE
Espone ed inquadra nel corretto contesto gli argomenti trattato. Usa con proprietà il linguaggio scientifico anche in lingua	9-10	Applica i procedimenti risolutivi in modo corretto e approfondito. Utilizza in modo completamente autonomo i software di laboratorio. Usa i formalismi dell'informatica in modo corretto.	9-10	Utilizza in modo completamente autonomo le strategie del pensiero razionale per risolvere problemi elaborando le opportune soluzioni dal punto di vista informatico.	Eccellente/ottimo
Espone in modo corretto gli argomenti trattati ed utilizza il lessico in modo appropriato anche in lingua	8	Utilizza in modo autonomo i software di laboratorio.	8	Utilizza in modo autonomo le strategie del pensiero razionale per risolvere problemi	buono
		Individua in modo autonomo le soluzioni ai problemi proposti e le realizza dal punto di vista		elaborando le opportune soluzioni dal punto di vista informatico.	

		informatico.			
Espone in modo corretto gli argomenti trattati ed utilizza il lessico in modo accettabile anche in lingua	7	Utilizza in modo adeguato i software di laboratorio. Individua con discreta autonomia le soluzioni ai problemi proposti e le realizza dal punto di vista informatico.	7	Utilizza in modo corretto le strategie del pensiero razionale per risolvere problemi ed è in grado di elaborare in modo autonomo le opportune soluzioni dal punto di vista informatico.	discreto
Coglie gli elementi di base dell'argomento trattato. Qualche incertezza nell'uso del linguaggio tecnico	6	Utilizza in modo parzialmente adeguato i software di laboratorio. Individua con parziale autonomia le soluzioni ai problemi proposti e le realizza dal punto di vista informatico.	6	Utilizza in modo parzialmente autonomo le strategie del pensiero razionale per risolvere problemi elaborando in modo quasi autonomo le opportune soluzioni dal punto di vista informatico.	sufficiente
Conosce in modo frammentario l'argomento trattato. Usa il linguaggio tecnico in modo incerto	5	Utilizza in modo non completamente adeguato i software di laboratorio. Individua con difficoltà le soluzioni ai problemi proposti e le realizza dal punto di vista informatico.	5	Utilizza in modo improprio le strategie del pensiero razionale per risolvere problemi e elabora con errori le opportune soluzioni dal punto di vista informatico	mediocre
Conosce in modo lacunoso l'argomento trattato. Gravi improprietà nell'uso del linguaggio tecnico	4	Utilizza in modo inadeguato i software di laboratorio. Non è in grado di individuare le soluzioni ai problemi proposti e li implementa con gravi errori.	4	Utilizza in modo improprio le strategie del pensiero razionale per risolvere problemi e elabora con gravi errori le opportune soluzioni dal punto di vista informatico	insufficiente
Conosce in modo gravemente lacunoso l'argomento trattato. Gravi improprietà nell'uso del linguaggio tecnico	2-3	Ignora completamente il funzionamento dei software di laboratorio. Non è in grado di individuare le soluzioni ai	2-3	Non è in grado di utilizzare le strategie del pensiero razionale per risolvere problemi e elabora con gravi errori le opportune soluzioni dal punto di vista informatico	Gravemente insufficiente

	problemi proposti e non è nemmeno in grado di implementarli.		
--	---	--	--

problemi proposti
e non è nemmeno
in grado di
implementarli.