IIS “G. Cigna G. Baruffi F. Garelli”	**Programmazione svolta**

Materia	Informatica

Rif. Programmazione dipartimentale	Segue la “Progr. Del Dipartimento di Matematica e Informatica *sede: “G.Baruffi””*
Triennio	Amministrazione Finanza e Marketing Articolazione Sistemi Informativi Aziendali
Classe	Classe 3^A SIA
Libro	Titolo: INPUT Corso di Informatica – **secondo biennio** Editore: sanoma linx Autori: Barbero, Rolfo, Rossi ISBN: 9788893797207

Introduzione all'informatica : Storia delle Architetture e dei Sistemi operativi.
CONTENUTO : 1. Evoluzione architetturale: da strumenti di ausilio al calcolo (Abaco), connessi all'adozione di un sistema di rappresentazione dei numeri posizionale additivo (meccanizzazione del calcolo, algoritmi delle operazioni), a macchine per il calcolo aritmetico (Pascal, Leibniz), fino all'idea di una macchina programmabile (Babbage e Ada Byron) 2. La Logica Matematica Proposta da G. Boole 3. Architettura di Von Neumann. 4. Evoluzione hardware: relè, valvola termoionica, transistor come elementi circuitali per implementare la logica e l'aritmetica binaria.

INTRODUZIONE ALLA PROGRAMMAZIONE

CONTENUTO:
• Problema Algoritmico e Algoritmo Risolvente
• Possibile definizione di algoritmo
• Formalismi in cui fornire la specifica di un algoritmo (pro e contro) Flow-Chart – Linguaggio Naturale - pseudo codice – Linguaggio di programmazione (nozione di "programma")
• Analisi del problema (dati in input, dati in uscita Output, relazione tra input e output)
• Variabili e costanti (l'istruzione di assegnazione, tipo ,contenuto, identificatore di una variabile)
• I Diagrammi di flusso (simboli grafici e significato)
• Esercizi attraverso Diagrammi di flusso
• Paradigma di programmazione Imperativo spaghetti coding e Paradigma di programmazione imperativo strutturato (Teorema Di Bohm Jacopini)
  • Gli Schemi di composizione fondamentali – Programmazione strutturata (sequenza-selezione-ripetizione)
  • Approfondimento sui cicli (condizionale, Post condizionale,, Ciclo For)

Le Variabili Strutturate e La loro Manipolazione
CONTENUTO: • Variabili semplici e variabili strutturate • Vettori e loro manipolazione • L'elaborazione dei vettori • Le matrici e loro manipolazione • L'elaborazione di matrici • Le stringhe di caratteri • Esercitazioni in laboratorio Mirate (Flowgorithm programmazione imperativa e strutturata, vettori e stringhe)

Sottoprogrammi E Ricorsione (paradigma procedurale)
• **CONTENUTO:** • L’approccio top-down e bottom-up • I sottoprogrammi • Parametri formali e attuali • Variabili Globali e Locali • Sperimentazione in laboratorio (Programmazione Procedurale in flowgorithm) • Contenuti eventuali: ◦ La ricorsione

Programmazione

CONTENUTO:

- Ambiente di sviluppo Integrato, presenti in laboratorio
- Il linguaggio Python e l'interprete (storia, motivazioni, evoluzione)
- The Python Standard Library , Documentazione Help online
- Caratteristiche del linguaggio:
    5. Operatori aritmetici,di confronto, logici (connettivi in e not)..
    6. Variabili, identificatori e tipi del linguaggio (linguaggio tipato dinamicamente→ non necessaria dichiarazione di tipo)
    7. L'output e l'input (le funzioni built-in input() print()).
    8. il controllo del flusso condizionale (if – else , elif)
    9. I cicli While (for visto quando si parla di stringhe e poi collezioni : stringhe,tuple,liste,dizionari)
    10. procedure e funzioni (type hints, valorizzazione dei parametri formali con valori attuali)
- implementazione di Programmi semplificati di esigenze reali anche con spunto al dominio economico aziendale.
-
    13. Le variabili strutturate e principali operazioni (scansione con for each, operatore in, operatore di sliceing, indicizzazione, len())
    14. Stringhe (metodi come split, replace, ecc)
    15. tuple, liste (append, remove, pop..)
    -

INDICAZIONI PER GLI STUDENTI CON INSUFFICIENZE E PER EVENTUALI ESAMI INTEGRATIVI O DI IDONEITÀ:

OBIETTIVI MINIMI;

1. Conoscere il significato di: algoritmo, programma, e problema algoritmico.
   Sapere spiegare le caratteristiche dei principali formalismi con cui è possibile descrivere un algoritmo.
2. Paradigma imperativo, strutturato . E Procedurale in teoria e con linguaggio python.
3. Saper interpretare codice python contenente costrutti della programmazione strutturata, procedurale, e che utilizzino colelzioni (stringhe, tuple, liste)

INDICAZIONI METODOLOGICHE PER LO STUDIO INDIVIDUALE ESTIVO;

- Per **TUTTI** gli allievi è **RICHIESTO SVOLGERE QUANTO SEGUE PER SETTEMBRE:**
  1. Ripassare la parte in "pdf" digitale del libro di testo "*richiami di programmazione dal primo biennio*" e "*linguaggi e strumenti per la programmazione*"
  2. Ripassare le prime 3 unità del libro (*cartacee*)
  3. Da portare **alla prima lezione:**
     1. su fogli ad anelli perché dovranno poi essere messi su quaderno **cartaceo di IV** leggere, comprendere, quindi produrre sintesi strutturata delle **U4, U5, U6.**
     2. Anche attraverso le conoscenze dell'U6 (utilizzo dei file) e delle precedenti, provare a realizzare un programma che consenta di gestire oggetti di un magazzino: il programma avviato presenta un menu con opzioni per : cercare un oggetto preferibilmente con un match parziale sul campo nome, aggiungere o modificare la quantità associata a un oggetto, salvare (su file), uscire. *Strutturare il codice in funzioni , usare commenti e type hints*. Si richiede file sorgente.

TIPOLOGIA DI PROVE CHE DOVRANNO SOSTENERE :

le prove (calendarizzate la stessa giornata e ragionevolmente in sequenza) consteranno in:

- **prova pratica:** richiesta di implementazione di codice python (1 esercizio di produzione, 2 di completamento, 2
- esercizio di stabilire l'output di un programma **SI VEDA ALLEGATO IN CALCE**)
- **Prova orale:** discussione dell'elaborato 15 min

I Docenti incaricati

I RAPPRESENTANTI DEGLI STUDENTI:

allegato alla programmazione svolta 1/4

PARTE 1: Completamento di codice (Produzione parziale)

Esercizio 1 (Stringhe e Funzioni) *Richiesta:* Completa la funzione censura_parola che riceve una stringa (una frase) e una parola da censurare. La funzione deve restituire la frase in cui la parola specificata è sostituita da una serie di asterischi * lunghi quanto la parola stessa.

Python

```
def censura_parola(frase, parola_da_nascondere):
  # Separa la frase in una lista di parole
  parole = frase._________________ # [COMPLETA QUI]

  risultato = []
  for p in parole:
    if p == parola_da_nascondere:
      # Crea una stringa di asterischi della stessa lunghezza
      asterischi = "*" * _________________ # [COMPLETA QUI]
      risultato.append(asterischi)
    else:
      risultato.append(p)

  # Ricompone la frase unendo le parole con uno spazio
  frase_censurata = " "._____________(risultato) # [COMPLETA QUI]
  return frase_censurata
```

Esercizio 2 (Liste e Tuple) *Richiesta:* Completa la funzione filtra_voti che riceve una lista di tuple. Ogni tupla contiene (nome_studente, voto). La funzione deve restituire una nuova lista contenente solo i nomi degli studenti che hanno la sufficienza (voto maggiore o uguale a 6).

allegato alla programmazione svolta 2/4

Python
```
def filtra_voti(registro):
  promossi = []
  # Itera su ogni tupla presente nella lista registro
  for studente in registro:
    # Estrai il nome e il voto dalla tupla
    nome = studente[0]
    voto = _________________ # [COMPLETA QUI]

    if voto >= 6:
      # Aggiungi il nome alla lista dei promossi
      promossi._________________(nome) # [COMPLETA QUI]

  return _________________ # [COMPLETA QUI]
```

PARTE 2: Interpretazione del codice

Esercizio 3 *Richiesta:* Analizza il seguente codice sorgente. Indica cosa viene stampato a video al termine dell'esecuzione e spiega brevemente il comportamento della funzione mistero.

Python
```
def mistero(sequenza):
  risultato = []
  for elemento in sequenza:
    if elemento not in risultato:
      risultato.append(elemento)
```

```
    return risultato


valori = [1, 2, 2, 3, 4, 4, 1, 5]
output = mistero(valori)
print(output)
```

Spazio per la risposta dello studente:

- Output stampato: _____
- Spiegazione (Cosa fa la funzione?): _____

Esercizio 4 *Richiesta:* Analizza il seguente codice incentrato sul passaggio dei parametri e sullo scope delle variabili. Indica cosa viene stampato a video e spiega perché il valore della variabile X nel programma principale non cambia.

Python

```
def modifica_dati(A, B):
    A = A + 10
    B.append(99)
    print("Dentro la funzione:", A, B)


X = 5
Y = [1, 2, 3]


modifica_dati(X, Y)
print("Nel main:", X, Y)
```

allegato alla programmazione svolta 4/4

Spazio per la risposta dello studente:

- Output della prima riga (Dentro la funzione): _____
- Output della seconda riga (Nel main): _____
- Spiegazione del comportamento di X e Y: _____

PARTE 3: Produzione di codice

Esercizio 5 *Richiesta:* Scrivi una funzione chiamata analizza_testo che riceve come parametro una stringa (un testo). La funzione deve calcolare e restituire una **tupla** contenente tre elementi:

1. Il numero totale di caratteri del testo (inclusi gli spazi).
2. Il numero di parole presenti nel testo.
3. La prima parola del testo. *Nota per lo studente: Ricordati di isolare le parole usando i metodi delle stringhe opportuni.*

Esercizio 6 *Richiesta:* Un negozio memorizza i prezzi dei suoi prodotti in una lista di numeri float chiamata listino. Scrivi una funzione chiamata applica_sconto che accetta due parametri: la lista listino e una percentuale di sconto (un numero intero, es. 20 per il 20%). La funzione deve creare e restituire una **nuova lista** con i prezzi scontati, arrotondati a due cifre decimali. Il listino originale non deve essere modificato. *Esempio:* applica_sconto([10.0, 50.0, 100.0], 10) deve restituire [9.0, 45.0, 90.0].