

# ANALISI SARTORIA

## Programma:

Realizzare un programma in linguaggio **C** per la gestione di una **sartoria**, focalizzato sul controllo e l'amministrazione dei **rotoli di tessuto** e dei **progetti di produzione**.

Il sistema deve consentire l'inserimento, la modifica e l'eliminazione di tessuti e progetti, la gestione dell'inventario secondo una logica **FIFO (First In, First Out)** e il calcolo dei costi e dell'usura dei materiali.

## Ipotesi:

- Ogni rotolo ha un costo e quando finisce viene comprato con le stesse dimensioni.
- L'usura dei rotoli aumenta ad ogni azione eseguita con essi, fino a quando non diventano inutilizzabili (50%).
- La sartoria ha un budget, è possibile andare in bancarotta se non si bilanciano spese e guadagni.
- Gli scarti vengono contati come pezzi (int) ed è possibile creare dei "*mini-progetti*" che utilizzino tot scarti.

## Struttura dati:

Ogni **rotolo** ha le sue determinate caratteristiche:

- Tipo di tessuto
- Colore
- Fantasia
- Lunghezza
- Larghezza
- Codice di fornitura
- Costo
- Usura (in %)

Mentre invece l'inventario sarà composto da **schede**:

- Codice rotolo
- Fornitore
- Rotolo
- Data di acquisto (Utilizzata nella FIFO)
- Quantità disponibile
- Utilizzo previsto

I progetti avranno una struttura dedicata che sarà composta da:

- Nome progetto
- Rotolo richiesto

- Metraggio richiesto
- Costo approssimato

## Requisiti:

### Requisiti Must:

1. Gestione tessuto (inserimento, modifica, eliminazione)
2. Gestione progetti (nuovo progetto, memorizzazione, calcolo costi, taglio)
3. Gestione inventario (controllo tessuti, verifica stato tessuti, rotazione scorta)

### Requisiti Should:

4. Implementazione Usura (Applica l'usura ad ogni taglio effettuato )
5. Salvataggio e Lettura dati ( Verrà usato un file per l'inventario e uno per i progetti)
6. Calcolo costi progetti ( Verranno calcolati costi per i progetti )

### Requisiti May:

7. Gestione budget ( entrate e uscite in base ai progetti )

## Constanti:

### MAXSTRING 30

- Dimensione massima delle stringhe

### MAXTESSUTI 100

- Numero massimo di schede con i relativi tessuti

### MAXPROGETTI 20

- Numero massimo di progetti

### BUDGETINIZIALE 1000

- Budget iniziale

### FILEINVENTARIO "Inventario.txt"

- Nome del file usato per salvare l'inventario

### FILEPROGETTI "Progetti.txt"

- Nome del file usato per salvare i progetti

## Structs:

Strutture dati:

```
struct rotolo{
    char tipo_tessuto[MAXSTRING];
    char colore[MAXSTRING];
    char fantasia[MAXSTRING];
    float lunghezza;
    float larghezza;
    char codice_fornitura[MAXSTRING];
    float costo;
    float usura;
};
```

```
struct data{
    int g;
    int m;
    int a;
```

```

};

struct scheda{
    char codice_rotolo[MAXSTRING];
    char fornitore[MAXSTRING];
    struct rotolo rot;
    struct data data_acquisto;
    float quantita_disponibile;
    float utilizzo_previsto;
    int scarti_utilizzabili;
} inventario[MAXTESSUTI];

struct progetto{
    char nome_progetto[MAXSTRING];
    char rotolo_richiesto[MAXSTRING];
    float metraggio_richiesto;
    float costo_approssimato;
    int mini; // se è un mini progetto verranno usati gli scarti
    float scarti_richiesti;
} progetti[MAXPROGETTI];

```

## Funzioni:

```

int menu(); // Stampa il menu
int nuovoRotolo(int*); // Aggiunge un nuovo rotolo
int modificaRotolo(int); // Modifica un rotolo esistente
int eliminaRotolo(int*); // Elimina un rotolo (azzerla la scheda e diminuisce la dimensione logica)
int nuovoProgetto(int*); // Aggiunge un nuovo progetto
int modificaProgetto(int); // Modifica un progetto
int eliminaProgetto(int*); // Elimina un progetto
float calcolaCostoProgetto(int); // Calcola il costo progetto (Ogni volta che il rotolo finisce lo
riacquista)
void mostraProgetti(int); // Stampa i progetti
int avviaTaglio(int*); // Avvia il taglio (rimuove i progetti in attesa)
void mostraTessuti(int); // Mostra i tessuti
int controlloTessuti(int); // Controlla i tessuti con usura troppo alta e ne propone la sostituzione
int rotazioneScorte(int); // Ruota le scorte
float aumentoUsura(); // Aumenta l'usura (a ogni azione o a caso. ancora da decidere...)
void salvaInventario(int,int); // Sotto forma di file .txt
void caricaInventario(int*,int*); // e salva sia tessuti che progetti

```

## Variabili:

Lista variabili (potrebbe cambiare durante la fase di sviluppo):

Dove	Nome	Cosa	Tipo	Valore
Globale	Finv	file usato per salvare l'inventario	FILE *	-
Globale	FProg	file usato per salvare i progetti	FILE *	-
Globale	budget	budget della sartoria	float	1000
main	RCount	tiene conto di quanti rotoli ci sono	int	0
main	PCount	tiene conto di quanti progetti ci sono	int	0
main	scelta	scelta effettuata in <code>menu()</code>	int	-
main	val	usata per leggere i dati per poi convertirli	char[ ]	10
main	err	riceve i codici di errore dalle funzioni	int	-
main	filter	usata come filtro in alcune funzioni	char[ ]	30
menu()	s1	variabile in <code>menu()</code> : prima scelta	int	-
menu()	s2	variabile in <code>menu()</code> : seconda scelta	int	-

## Struttura file:

La struttura dei file sarà organizzata in questo modo:

Dati

Rcount=1

Pcount = 1

inventario[0]:

- rotolo1
- Rotoli&Co
- dati rotolo...
- 10/08/2024
- 10 M^2
- 7
- 0

progetti[0]:

- progetto1
- rotolo1
- 5
- 30€
- 0
- 0

File:

```
1 1
rotolo1 Rotoli&Co [dati rotolo] 10 08 2024 10 7 0
progetto1 rotolo1 5 30 0
```

Le variabili verranno separate da spazi e successivamente lette dalla funzione `fscanf()`