**Cognome Nome**

**Esercitazione del 25/03/2020 Classe 4AS A.S. 2019/20**

**Argomenti dell’esercitazione:**

* Argc/Argv
* File di Testo
* Funzioni con Passaggio di Parametri

**Attività:**

* Produzione Codice Richiesto dalla Traccia
* Condivisione in area Personale GitLab
* Consegna in area Classroom della traccia con la compilazione delle parti indicate in fondo.

Esercizio: “Minuti lavorati”

Un’azienda ha dotato i propri dipendenti di un sensore wireless che emette un codice numerico ogni volta che un dipendente attraversa la porta d’ingresso/uscita dell’azienda o ne transita nelle vicinanze. L’azienda ha meno di 1000 dipendenti. Ad ogni attraversamento, il sensore registra ora e minuti del passaggio, insieme al codice del dipendente (un codice alfanumerico di max 10 caratteri).

Si desidera sviluppare un programma in linguaggio C per il calcolo delle ore lavorative dei dipendenti dell’azienda. Il programma riceve sulla linea di comando un primo parametro, che rappresenta il nome del file contenente gli attraversamenti, ed un secondo parametro (opzionale), che rappresenta il codice numerico di un dipendente.

Il file e relativo ai passaggi di una sola giornata, ed` e composto da una serie di righe, ciascuna delle` quali corrisponde ad un passaggio, ed e composta da tre campi:`

**ora minuti codice\_dipendente**

Se il programma viene invocato con un due parametri sulla linea di comando (vi è il codice del dipendente), allora dovrà stampare, per il dipendente specificato, il numero totale di minuti lavorati. Per determinare il numero di minuti lavorati occorre confrontare l’orario del *primo* passaggio con l’orario dell’*ultimo* passaggio per quel dipendente.

Se invece il programma viene invocato con un solo parametro sulla linea di comando (il codice del dipendente e assente), allora il programma dovrà stampare il numero totale di dipendenti` *diversi* che hanno lavorato in quel giorno (ossia che sono passati almeno una volta dalla porta).

Ad esempio, dato il seguente file di testo passaggi.txt:

8 30 abc222

8 30 abc123

* 31 azx112
* 10 abc123

12 10 abc123

il programma (chiamato orario.c) si dovrà comportare nel modo seguente:`

c:> orario passaggi.txt

Ci sono 3 dipendenti diversi.

c:> orario passaggi.txt abc123

Il dipendente abc123 ha lavorato per 220 minuti.

**SOLUZIONE PROPOSTA**

**Analisi**

La richiesta posta, prevede lo sviluppo di un algoritmo che segna il calcolo dei minuti di lavoro svolti da un dipendente di un’azienda. I dati utili al calcolo del tempo lavorativo (orario di entrata, orario di uscita e codice di riconoscimento del dipendete), saranno prelevati da un file di testo.

Inserendo da linea di comando uno dei codici dipendente, il programma dovrà mostrare a video il totale dei minuti che quel dipendente ha lavorato. Qualora quel dipendente assente, il programma dovrà visualizzare la lista completa dei dipendenti che quel giorno erano presenti in azienda.

**Descrizione Funzioni**

**int leggi(char\* nomefile, struct dipendenti\* accessi)**

Scopo:

leggere i passaggi dal file e memorizzarli nella **struct** accessi

Descrizione parametri:

**char\* nomefile**: puntatore all'array di caratteri contenente il nome del file da aprire.

**struct dipendenti\* accessi**: puntatore alla struttura che conterra' i passaggi letti da file.

Descrizione output prodotto:

**return 1:** OK

**return 0:** ERRORE

**int opzione1(struct dipendenti accessi)**

Scopo:

restituire il numero totale di dipendenti DIVERSI entrati nell'azienda

Descrizione Parametri:

**struct dipendenti accessi**: copia della struttura contenente i passaggi letti da file

Descrizione output prodotto:

**return cont** (numero dipendenti diversi trovati)

nel main -> printf "in totale ci sono x dipendenti" **x = cont**

**void opzione2(char matricola[], struct dipendenti accessi)**

Scopo:

restituire il numero totale di minuti in cui il dipendente scelto e' stato all'interno dell'azienda

Descrizione parametri:

**char matricola[]**: nome della matricola (passata come parametro al main)

**struct dipendenti accesso**: copia della struttura contenente i passaggi letti da file.

Descrizione output:

**OK**: printf "il dipendente abc123 ha lavorato 200 minuti"

**ERR**: printf "il dipendente ha effettuato solo un passaggio"

**ERR**: printf "dipendente non trovato"

**Descrizione Main**

**int argc** = contatore che ci dice il numero di parametri passati al main.

**char\*\* argv** = matrice di caratteri:

**argv[1][]** = nome del file da aprire

**argv[2][]** = nome della matrice

**if(arcg == 2)** -> se e' stato passato un solo parametro (oltre al nome del programma)

**if(argc == 3)** -> se sono stati passati due parametri (oltre al nome del programma)

il ciclo **for** serve semplicemente a trasformare la matrice in minuscolo, per evitare errori banali.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Allegare Codice Prodotto**

GitHub: [**https://github.com/piemo35/C/tree/master/ore\_dipendenti**](https://github.com/piemo35/C/tree/master/ore_dipendenti)

GitLab: [**https://gitlab.com/piemo-c/es\_orario\_dipendenti**](https://gitlab.com/piemo-c/es_orario_dipendenti)