Testo esercizio:

Si supponga di avere a disposizione un vettore contenente l'elenco degli alunni iscritti al Politecnico, con la relativa data di nascita.
Ogni nome è separato dalla data dal carattere ':'.
La data è espressa come ggmmaaaa.

Risolvere i seguenti problemi::

1. Scrivere una funzione che restituisce la data dinascita
   di un determinato studente.

2. Scrivere una funzione che elimina tutti gli studenti
   nati in una determinata data

3. Scrivere una funzione che conta il numero degli studenti
   nati in ogni mese dell'anno.

4. Scrivere una funzione che determina gli studenti più
   vecchi.

Bisogna produrre:

- La descrizione precisa e completa della struttura dati utilizzata. (p.ti. 2)
- Per ogni problema la codifica in c e il flow chart della
  funzione principale. (p.ti 4)

Esistono diverse strutture dati che possono essere utilizzate per la rappresentazione dei dati.
Si può utilizzare un vettore di puntatori a stringhe oppure una matrice di caratteri.

Nel seguito utilizzerò una matrice di caratteri così definita:

```
#define N 1000
#define SIZE 256

typedef char Studente[SIZE];
typedef Nome Elenco[N];
```

La stringa vuota verrà utilizzata come segnale di fine elenco.

Inoltre suppongo di avere a disposizione alcune funzioni che sicuramente mi serviranno per la soluzione del problema.

```
int myStrcmp(Studente e, char * nome);

   confronta un elemento della mia matrice con una stringa
   contenente un nome di persona. Restituisce 0 se uguali,
   un numero negativo se il nome in e è minore di nome,
   un numero positio se il nome in e è maggiore.

void dataCpy(char * data, Studente e);

        copia la data contenuta in e in data

int dataCmp(Studente e, char *data);

   confronta una data contenuta in e con data.
   restituisce 0 se sono due date uguali,
   numero negativo se la data contenuta in e precede data
   un numero positivo se la data contenuta in e segue data.

int anno(Studente e);

        restituisce l'anno di nascita, come intero

int mese(Studente e);

        restituisce il mese di nascita, come intero

int giorno(Studente e);

        restituisce il giorno di nascita, come intero
```

problema 1.

La funzione che risolve questo problema ha il seguente prototipo.

```
     void dataNascita(Elenco e, char nome[], char data[]);
```

Deve cercare in e se esiste uno studente che si chiama

nome
e copiare la data di nascita in data.
Se lo studente non esiste restituisce una data con '\0'
nella prima posizione.

```
void dataNascita(Elenco e, char nome[], char data[]){
        int i;

        for(i=0; e[i][0]!='\0'; ++i)
                if(myStrcmp(e[i], nome) == 0){
                        dataCpy(data, e[i]);
                        return;
                }

        data[0]='\0';  /* studente non trovato  */
}
```

problema 2

La funzione che risolve questo problema ha il seguente prototipo.

```
        void cancella(Elenco e, char data[]);
```

Per eliminare si usa il seguente algoritmo: si copiano tutti gli studenti che si vogliono tenere.

```
void cancella(Elenco e, char data[]){
        int i, k;

        for(i=0, k=0; e[i][0]!='\0'; ++i)
                if( ! dataCmp(e[i], data ){
                        strcpy(e[k], e[i]);
                        k++;
                }

        e[k][0] = '\0'; /* si rimette il tappo alla fine
*/
}
```

problema 3

La funzione che risolve questo problema ha il seguente prototipo.

```
        void contaNelMese(Elenco e, int conta[]);
```

La funzione ha in ingresso l'elenco e restituisce un vettore di contatori che è già inizializzato con 0, contenente nella cella 1 i nati in gennaio, nella cella 2 i nati in febbraio, e così via

```
void contaNelMese(Elenco e, int conta[]){
        int i;

        for(i=0; e[i][0]|='\0'; ++i)
                conta[mese(e[i])]++;
}
```

problema 4

La funzione che risolve questo problema ha il seguente prototipo.

```
        void piuVecchi(Elenco e, Elenco vecchi);
```

copia in vecchi tutti gli studenti che sono più vecchi.
E' un problema di ricerca del minimo.
Trovata la data del più vecchio, cioè chi è nato prima, copia in vecchi tutti gli studenti nati in quella determinata data.

```
void piuVecchi(Elenco e, Elenco vecchi){
        char d[9];
        int i, k;

        cercaDataVecchio(e, d);  /*trova data più
vecchio*/

        for(i=0, k=0;e[i][0] != '\0'; ++i)
                if(dataCmp(e[i], d) == 0){
                        strcpy(vecchi[k], e[i]);
                        k++;
                }

        vecchi[k][0]='\0'; /* tappo a fine elenco  */
}
```

la seguente funzione cerca la data più piccola.

```
void cercaDataVecchio(Elenco e, char d[]){
        int i;

        dataCpy(d, e[0]);

        for(i=0; e[i][0]!='\0'; ++i)
                if( dataCmp(e[i], d) < 0 )
                        /* se la data in e[i] precede d
*/
                        dataCpy(d, e[i]);
}
```