

| | | | | | | | | | |
|---|---|---------|----------------|--|--|--|--|--|--|
|  | Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione FONDAMENTI DI INFORMATICA Appello 28 Febbraio 2017 | | COGNOME E NOME | | | | | | |
| | RIGA | COLONNA | MATRICOLA | | | | | | |
| <div style="text-align: right;">Spazio riservato ai docenti</div> <table border="1" style="margin-left: auto;"> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> | | | | | | | | | |
| | | | | | | | | | |

- Il presente plico contiene 4 esercizi e **deve essere debitamente compilato con cognome e nome, numero di matricola.**
- Il tempo a disposizione è di 1 ore e 20 minuti.
- Non separate questi fogli. Scrivete la soluzione solo sui fogli distribuiti, utilizzando il retro delle pagine in caso di necessità. Cancellate le parti di brutta con un tratto di penna.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- È possibile scrivere a matita (e non occorre ricalcare al momento della consegna!).
- **È vietato utilizzare telefoni o pc.** Chi tenti di farlo vedrà annullata la sua prova.
- È ammessa la consultazione di libri e appunti.
- **Qualsiasi tentativo di comunicare con altri studenti comporta l'espulsione dall'aula.**
- È possibile ritirarsi senza penalità.
- **Non è possibile lasciare l'aula conservando il tema della prova in corso.**

Esercizio 1 (12 punti)

Un'azienda registra su **file testuali** gli ingressi e le uscite dei propri dipendenti. Per ogni dipendente, l'ingresso e l'uscita di una giornata lavorativa sono organizzati su una riga formattata nel seguente modo:

```
matricola giorno mese anno ingresso uscita
```

dove *matricola* è un codice numerico di 6 cifre che identifica ogni dipendente (un numero a 6 cifre); *giorno*, *mese* ed *anno* rappresentano la data della registrazione e sono rappresentati in maniera numerica; *ingresso* ed *uscita* sono gli orari di ingresso e uscita, rappresentati come numero di minuti trascorsi dalla mezzanotte precedente (ad esempio, le ore 7:30 corrispondono a 450 minuti, infatti $7 * 60 + 30 = 450$).

Ad esempio la seguente riga:

```
123456 28 2 2017 450 1050
```

indica che il dipendente con matricola 123456, il giorno 28/2/2017 è entrato alle ore 7:30 ($7*60+30 = 450$) ed è uscito alle ore 17:30 ($17*60+30 = 1050$).

Implementare la seguente funzione in C:

```
int max(int mese, int anno, const char nome[]);
```

che riceve in ingresso mese, anno e il nome del file su cui sono memorizzati ingressi ed uscite; la funzione deve quindi ritornare la matricola del dipendente che nel mese e anno ricevuto in ingresso ha passato più tempo al lavoro (il tempo al lavoro viene calcolato come la differenza fra uscita ed entrata analizzando il file di registro passato come parametro).

Note. Nell'implementare la funzione max, si può ipotizzare che **vi siano al massimo 100 dipendenti diversi (e almeno un dipendente)**. **Le righe nel file del registro ingressi/uscita sono ordinate per numero di matricola.** Non si può invece supporre che il file sia ordinato rispetto alle date degli accessi.

Soluzione

```
int max(int mese, int anno, const char nome[])
{
    FILE *f;
    int matricole[100], tempo[100], k=-1, i, imax=0;

    f = fopen(nome, "r");

    while (feof(f)==0)
    {
        int matricola, g, m, a, in, out;
        fscanf(f, "%d%d%d%d%d", &matricola, &g, &m, &a, &in, &out);

        if ( k==-1 || matricola != matricole[k] )
        {
            k++;
            matricole[k] = matricola;
            tempo[k] = 0;
        }
        if (m == mese && a == anno)
            tempo[k] = tempo[k] + out-in;
    }

    for (i=1; i<k; i++)
        if (tempo[i] > tempo[imax])
            imax = i;

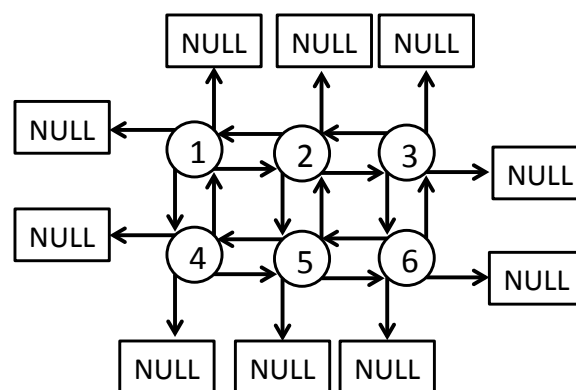
    return matricole[imax];
}
```

Esercizio 2 (12 punti)

Si consideri la seguente struttura dati dinamica che permette di rappresentare una *griglia*:

```
struct griglia
{
    int dato;
    struct griglia *su;
    struct griglia *giu;
    struct griglia *dx;
    struct griglia *sx;
};
```

Utilizzando opportunamente la struttura dati qui sopra è possibile realizzare una griglia rettangolare di elementi organizzati in righe e colonne, dove per indicare l'assenza di un elemento sulla riga/colonna successiva si utilizza il valore NULL. Ad esempio una griglia di 2 righe e tre colonne può essere rappresentata con la precedente struttura dati nel seguente modo:



Implementare la seguente funzione C che riceve in ingresso un puntatore al primo (a partire da sinistra) elemento di una riga di una griglia ed elimina dalla struttura dati l'intera riga corrispondente:

```
void elimina_riga(struct griglia *riga);
```

Note. Si assuma che tutte le righe di una griglia abbiano sempre lo stesso numero di elementi. La funzione implementata deve anche liberare la memoria allocata agli elementi cancellati. Non è possibile assumere che esista sempre una riga che precede né una riga che segue quella che deve eliminare.

Soluzione

```
void elimina_riga(struct griglia *riga)
{
    while (riga != NULL)
    {
        if (riga->su != NULL) //se esiste una riga precedente
            riga->su->giu = riga->giu; //collega la riga precedente alla successiva

        if (riga->giu != NULL) //se esiste la riga successiva
            riga->giu->su = riga->su; //collega la riga successiva alla precedente

        struct griglia *temp = riga; //salva temporaneamente l'elemento corrente
        riga = riga->dx; //si sposta in avanti
        free(temp); //libera la memoria
    }
}
```

Si consideri il numero negativo e razionale -216.25
Quanto vale la sua codifica in virgola mobile?

Soluzione



Rispondere alle seguenti domande giustificando le risposte:

- A. Quale è la struttura dell'indirizzo virtuale e di quello fisico, e la lunghezza dei campi che li costituiscono?
- B. Un consulente afferma che essendo la memoria fisica e la memoria virtuale della medesima dimensione, non potranno mai esserci pagine **non residenti** in memoria e quindi non è necessario prevedere uno spazio sul disco fisso per ospitare le pagine non residenti. Siete d'accordo? Argomentare in maniera adeguata la propria risposta.

Soluzione

- A. 16 Mbyte di memoria indirizzabile \rightarrow 24 bit di indirizzo
Pagina di 16Kbyte = 2^{14} byte \rightarrow 14 bit offset

Quindi si ha,

Indirizzo virtuale: NPV: 10 bit, offset: 14 bit

Indirizzo fisico: NPF: 10 bit, offset: 14 bit

- B. L'affermazione è errata, dal momento che la memoria fisica è condivisa da più processi, quindi anche se la memoria virtuale del **singolo** processo non può eccedere la dimensione della memoria fisica, l'insieme dei processi caricati in un dato momento potrebbero avere delle richieste di memoria che superano la dimensione della memoria fisica.