

UNIVERSITA' DI BERGAMO

ESAME DI INFORMATICA 12 CFU – Modulo di Programmazione (ING. INFORMATICA)

Prof. G. PSAILA

APPELLO DEL 21/02/2020

Per consegnare, si svolgano entrambi gli esercizi. Durata: 90 minuti.
Punteggio complessivo: 16 punti. Sufficienza: 9 punti.

Esercizio (10 punti)

Si consideri un programma per gestire la raccolta delle informazioni relative agli abbonamenti ad un parcheggio. Un abbonamento è descritto da un tipo strutturato denominato `ABBONAMENTO`, i cui campi sono un codice di 10 caratteri (che identifica univocamente l'abbonamento), la data di emissione (stringa nel formato internazionale `aaaa-mm-gg`), la data di scadenza, il numero di targa (stringa di 10 caratteri), l'importo residuo (numero in virgola mobile), il valore dell'ultimo addebito e la data dell'ultimo addebito..

Si definisca quindi la struttura dati per una lista dinamica dove il campo informativo del nodo è a sua volta basato sul tipo `ABBONAMENTO`.

Si scriva la funzione denominata `SpostaAbbonamenti` che riceve come parametri due liste di abbonamenti, denominate rispettivamente `lista1` e `lista2`, un vettore di stringhe (vettore di puntatori a carattere) denominato `elencoCodici`, ed un parametro `size` (numero intero) che indica quanti sono gli elementi nel vettore `elencoCodici`. Per ogni codice presente nel vettore `elencoCodici`, la funzione verifica se il codice è presente in `lista1`: se un abbonamento con il codice cercato non è presente in `lista1`, lo cerca in `lista2` e, se lo trova, copia l'abbonamento in questione in `lista1`.

La funzione restituisce il numero di abbonamenti copiati da `lista2` a `lista1`; in caso di errore, la funzione restituisce -1.

Domanda Teoria (6 punti)

Si consideri un sistema in cui per gli indirizzi di memoria vengono usati 24 bit e la memoria viene gestita con il sistema della paginazione con pagine da 1Kbyte e indirizzi logici. Si consideri il seguente indirizzo logico $l=000000000011100000101111$.

Se nella tabella delle pagine abbiamo le corrispondenze $pl \rightarrow pf$ (in base 10) $13 \rightarrow 10$, $14 \rightarrow 7$, $17 \rightarrow 53$, $20 \rightarrow 102$, qual è l'indirizzo fisico f su 24 bit corrispondente all'indirizzo logico l ?

Soluzione del programma

```
#include <cstring>
```

```
struct ABBONAMENTO
{
    char codice[11];
    char data_emissione[11];
    char data_scadenza[11];
    char targa[11];
    float importo_residuo;
    float ultimo_addebito;
    char data_ultimo_addebito[11];
};
```

```
struct NODO
{
    ABBONAMENTO info;
    NODO *next;
};
```

```
NODO *cerca(NODO *head, char codiceCercato[])
{
    NODO *p;

    for(p=head; p!= NULL; p=p->next)
    {
        if( strcmp(p->info.codice, codiceCercato)==0)
            return p;
    }
    return NULL;
}
```

```
int ins_testa(NODO *&head, NODO *pAbbonamento)
{
    NODO *t;

    t = new NODO;
    if( t == NULL)
        return 1;

    t->info = pAbbonamento->info;
    t->next = head;
    head = t;

    return 0;
}
```

```
int SpostaAbbonamenti( NODO *&lista1, NODO *lista2,
    char *elencoCodici[], int size)
{
    NODO *ricerca1;
    NODO *ricerca2;
```

```

int i;
int contatore=0;

for(i=0; i < size; i++)
{
    ricerca1 = cerca( lista1, elencoCodici[i] );
    if( ricerca1 == NULL)
    {
        ricerca2 = cerca(lista2, elencoCodici[i] );
        if( ricerca2 != NULL)
        {
            if( ins_testa(lista1, ricerca2)!=0)
                return -1;
            contatore++;
        }
    }
}
return contatore;
}

```

Soluzione teoria

Con pagine da 1 K Byte, cioè 1024 Byte, servono 10 bit per l'offset, perché $2^{10} = 1024$.
Quindi, i primi 14 bit dell'indirizzo sono il numero di pagina logica, mentre i restanti 10 sono l'offset

$l = 000000000001110 \ 0000101111$

Risulta pertanto che la pagina logica $pl = 000000000001110_2 = 14_{10}$.

Nella tabella delle pagine, la pagina fisica corrispondente è 7_{10} , che in binario su 14 bit è

$pf = 000000000000111_2$

L'indirizzo fisico si ottiene, quindi, giustapponendo i 14 bit del numero di pagina e i 10 bit dell'offset, cioè

$f = 000000000000111 \ 0000101111 = 000000000000111 \ 0000101111$