

# UNIVERSITA' DI BERGAMO

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

Prof. G. PSAILA

### APPELLO DEL 22/01/2021

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 le registrazioni degli esami sostenuti dagli studenti universitari. Una registrazione è descritta da un tipo strutturato denominato `REGISTRAZIONE`, i cui campi sono un codice di 10 caratteri (che identifica univocamente la registrazione), la data della registrazione (stringa nel formato internazionale `aaaa-mm-gg`), la data dell'appello, il numero di matricola dello studente (stringa di 10 caratteri), il codice dell'insegnamento e il codice del corso di laurea (entrambi stringhe di 10 caratteri) e il voto ottenuto (numero intero da 0 a 31, dove 31 indica la lode).

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

Si scriva la funzione denominata `MiglioriRipetizioni`, che riceve come parametri due liste di registrazioni denominate `lista1` e `lista2`, un vettore di codici di insegnamento (vettore di stringhe) denominato `codiciIns` e un parametro `size` che indica il numero di elementi nel vettore `codici_Ins`.

La funzione restituisce una nuova lista di registrazioni, gestita internamente tramite la variabile `listaOut`; una registrazione presente in `lista1` viene inserita in `listaOut` se il codice dell'insegnamento è presente in `codiciIns` e se vale una delle situazioni seguenti: a) non esiste in `lista2` una registrazione relativa allo stesso studente e allo stesso insegnamento; b) esiste in `lista2` una registrazione relativa allo stesso studente e allo stesso insegnamento, ma il voto ottenuto in quest'ultima è inferiore al voto ottenuto nella registrazione presente in `lista1`.

La funzione restituisce l'indirizzo della testa della nuova lista; la funzione restituisce `NULL` se non c'è nulla da restituire o in caso di errore.

N.B. si eviti la duplicazione di parti del codice del programma.

#### **Domanda Teoria (6 punti)**

Si consideri il carattere il cui codice UNICODE è `430010`. Si riporti l'encoding UTF-8 corrispondente, facendo vedere i passaggi effettuati.

## Soluzione Esercizio

```
struct REGISTRAZIONE
{
    char codice[11];
    char data_reg[11];
    char data_appello[11];
    char matricola[11];
    char cod_ins[11];
    char corso_laurea[11];
    int voto;
};

struct NODO
{
    REGISTRAZIONE dato;
    NODO *next;
};

int ins_testa(NODO *&head, REGISTRAZIONE dato)
{
    NODO *t;

    t = new NODO;
    if( t== NULL)
    {
        cout << "Memoria Esaurita";
        return 1;
    }

    t->dato = dato;
    t->next = head;
    head = t;

    return 0;
}

int CercaInsegnamento(char codice[], char *codici_Ins[], int size)
{
    int i;

    for(i=0; i < size; i++)
        if( strcmp(codice, codici_Ins[i])== 0)
            return 1;

    return 0;
}

int Verifica(NODO *riferimento, NODO *head)
{
    int trovato=0;
    NODO *p;

    p = head;
    while(p != NULL)
```

```

{
    if(strcmp(riferimento->dato.cod_ins, p->dato.cod_ins)==0
    && strcmp(riferimento->dato.matricola,
            p->dato.matricola)==0)
    {
        if( riferimento->dato.voto >p->dato.voto )
            return 1;
        else
            trovato = 1;
    }
    p = p->next;
}
return 1-trovato;
}
/* NOTA: trovato diventa 1 se la registrazione viene trovata ma
il voto supera quello della registrazione di riferimento.
Se si esce dal ciclo e una registrazione con voto superiore
a quella di riferimento viene trovata, si deve restituire 0.
Se si esce dal ciclo e nessuna registrazione corrispondente
esiste, si deve restituire 1 (condizione a)
*/

NODO *MiglioriRipetizioni(NODO *lista1, NODO*lista2,
                        char *codici_Ins[], int size)
{
    NODO *listaOut=NULL;

    NODO *p;
    int da_inserire;

    p = lista1;
    while(p != NULL)
    {
        if(CercaInsegnamento(p->dato.cod_ins, codici_Ins, size))
        {
            da_inserire = Verifica(p, lista2);
            if( da_inserire)
            {
                if( ins_testa(listaOut, p->dato)==1)
                    return NULL;
            }
        }
        p = p->next;
    }
    return listaOut;
}

```