Informatica A – 18-2-2020

**Cognome (IN STAMPATELLO)**

**Nome**

**Matricola o Codice Persona**

Istruzioni

1. Non separate questi fogli. Scrivete la soluzione **solo** **sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità.
2. **È possibile scrivere a matita** (e non ricalcare al momento della consegna).
3. Ogni parte non cancellata sarà considerata parte integrante della soluzione.
4. È **vietato** utilizzare **calcolatrici** o **telefoni**. Chi tenti di farlo vedrà **annullata** la sua prova.
5. Non è ammesso consultare **libri** o **appunti** o altro.
6. Qualsiasi **tentativo** di comunicare con altri studenti comporta **l’espulsione** dall’aula.
7. È possibile **ritirarsi senza penalità** lasciando il tema d’esame con nome e cognome.
8. Non è possibile lasciare l’aula conservando il tema della prova in corso.
9. Tempo a disposizione: 2 ore e 30 minuti

**Esercizio 1 ( 2 punti )**

**Esercizio 2 ( 4 punti )**

**Esercizio 3 ( 4 punti )**

**Esercizio 4 ( 10 punti )**

**Esercizio 5 ( 8 punti )**

**Esercizio 6 ( 4 punti )**

**Voto finale:**

Esercizio 1 ( 2 punti )

Si costruisca la tabella di verità della seguente espressione booleana.

( C or ( not B ) ) or ( ( not A ) or B)

Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = -101 e B = 118, li si converta, se ne calcoli la differenza (A-B) in complemento a due e si indichi se si genera riporto sulla colonna dei bit più significativi e se si verifica overflow.

Esercizio 2 ( 4 punti )

Lo schema sottostante descrive un portale per la pubblicazione e fruizione di video.

UTENTE (USERNAME, NOME, COGNOME, DATANASCITA, EMAIL)

VIDEO (IDVIDEO, TITOLO, USERNAMECREATOR, GENERE, DURATA, DATAPUBBLICAZIONE)

VISIONE (IDVIDEO, USERNAME, DATAORAVISIONE, MINUTIGUARDATI, VALUTAZIONE)

Scrivere una query che estrae nome e cognome degli utenti che non hanno mai visto un video.

Lo schema sottostante descrive un portale per la pubblicazione e fruizione di video.

UTENTE (USERNAME, NOME, COGNOME, DATANASCITA, EMAIL)

VIDEO (IDVIDEO, TITOLO, USERNAMECREATOR, GENERE, DURATA, DATAPUBBLICAZIONE)

VISIONE (IDVIDEO, USERNAME, DATAORAVISIONE, MINUTIGUARDATI, VALUTAZIONE)

Scrivere una query che estrae, per ogni genere, il titolo del video col maggior numero di visualizzazioni.

Esercizio 3 ( 4 punti )

Si consideri il seguente programma, completando l’inizializzazione del vettore V con la propria matricola (composta da 6 cifre) e si scriva nel riquadro sottostante l’output stampato a schermo.

void f(char \* str);

int main() {

char V[7] = "......"; // METTERE QUI LA VOSTRA MATRICOLA

f(V+3);

return 0;

}

void f(char \* str) {

printf("%c", \*str);

str=str+1;

if(strlen(str) == 0)

return;

else {

f(str);

printf("%c", \*str);

return;

}

}

>>>

Esercizio 4 ( 10 punti )

Sia una funzione analitica di due variabili e a valori reali. Si assuma che sia stata campionata in una matrice float **M**[100][100] tale per cui ogni posizione i,j della matrice contiene il valore di nelle coordinate :

**M**[i][j] =

Si scriva una funzione **estraiMassimiLocali** che prende in ingresso la matrice **M** e restituisce una lista contenente tutti i massimi locali di **M**. Si definisca preventivamente un opportuno tipo di dato **Nodo** che deve contenere, oltre alle coordinate del massimo locale, anche il valore della funzione.

Si scriva un frammento di codice per invocare **estraiMassimiLocali** sulla matrice **M,** assumendo che questa sia già stata popolata.

Un massimo locale è una coordinata (i,j) dove il valore di **M**[i][j] è strettamente maggiore di tutte le 8 coordinate (i,j) adiacenti.

Si suggerisce di scrivere una funzione ausiliaria **maxLoc** che restituisce il valore massimo tra tutti i valori adiacenti ad una posizione passata in ingresso.

Ricordatevi di gestire i bordi della matrice (in questi casi le caselle adiacenti sono meno di 8).

Esercizio 5 ( 8 punti )

Si consideri la seguente definizione di lista:

**typedef struct EL {**

**int dato;**

**struct EL \* next;**

**} nodo;**

**typedef nodo \* lista;**

Scrivere una funzione

**Lista Merge(Lista lista1, Lista lista2)**

che prese in input due liste ordinate per valori crescenti, fornisce in output una nuova lista ordinata contenente tutti gli elementi delle due liste (duplicati inclusi).

Esercizio **6** ( 4 punti )

Si consideri la seguente definizione di albero binario che contiene i dati relativi alla gerarchia aziendale di un’azienda in cui ogni persona è a capo di uno o due sottoposti:

**typedef struct node\_s {**

**char nome[100],cognome[100];**

**struct node\_s \* left, \*right;**

**} node\_t;**

**typedef node\_t \* tree;**

Scrivere una funzione

**void** **cognomeDiffuso(tree T, char str[])**

che preso in ingresso un albero e una stringa, inserisca nella stringa il cognome più diffuso nell’azienda (si supponga sia unico).