Informatica A

**Cognome Matricola**

**Nome Firma**

Istruzioni

1. 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** (o ripudiate) con un tratto di **penna**.
2. Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
3. **È possibile scrivere a matita** (e non occorre ricalcare al momento della consegna!).
4. È **vietato** utilizzare **calcolatrici,** **telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
5. È ammessa la consultazione di **libri** e **appunti**, purché con pacata discrezione e senza disturbare.
6. Qualsiasi **tentativo** di comunicare con altri studenti comporta **l’espulsione** dall’aula.
7. È possibile **ritirarsi senza penalità**.
8. Non è possibile lasciare l’aula conservando il tema della prova in corso.
9. Tempo a disposizione: **2h 30min**

Valore indicativo degli esercizi, voti parziali e voto finale:

**Esercizio 1 ( 2 punti )**

**Esercizio 2 ( 4 punti )**

**Esercizio 3 ( 4 punti )**

**Esercizio 4 ( 4 punti )**

**Totale: ( 14 punti )**

Esercizio 1 (2 punti)

Si costruisca la tabella di verità della seguente espressione booleana.

**(not A) or ( B and C ) or ( A or (C or B) )**

Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = 11dec e B = –111dec, li si converta, se ne calcolino la somma (A+B) e 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)

Si dica cosa stampa il seguente codice:

#define N 50

#include<stdio.h>

#include<conio.h>

int f1(char \* v1, char \* v2);

int f2(char \* v1, char \* v2);

int f3(char \* v1, char \* v2);

int main(){

char z[N]="melo",w[N]="melograno";

printf("%d,%d,%d\n",f1(z,w),f1(z,z),f1(w,w));

printf("%d,%d,%d,%d\n",f2(z,w),f2(w,z),f2(z,z),f2(w,w));

printf("%d,%d,%d,%d\n",f3(z,w),f3(w,z),f3(z,z),f3(w,w));

getch();

return 1;

}

int f1(char \* v1, char \* v2){

int i=0;

while(1){

if(v1[i]=='\0' || v1[i]=='\0')

return i;

i++;

}

return 0;

}

int f2(char \* v1, char \* v2){

int i=0;

while(v1[i]!='\0'){

if(v1[i]!=v2[i] || v2[i]=='\0')

return i;

i++;

}

return 0;

}

int f3(char \* v1, char \* v2){

int i=0;

while(v1[i]!='\0' && v2[i]!='\0' && v1[i]==v2[i])

i++;

if(i==f1(v1,v2) && i==f1(v1,v1) && i==f1(v2,v2))

return 2;

else if(i==f1(v1,v2))

return 1;

else

return 0;

}

Esercizio 3 (4 punti)

Si considerino le seguenti dichiarazioni di tipi che definiscono le strutture dati per gestire ripetizioni private.

typedef struct {

int giorno, mese, anno;

} data;

typedef struct {

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

data DataNascita;

} persona;

typedef struct {

persona docente, studente;

data d;

} lezione

typedef lezione ripetizioni[1000];

Si definisca una funzione di prototipo

int f(ripetizioni r);

che restituisce il numero di docenti che hanno dato ripetizioni a uno studente il giorno in cui lo studente compiva 18 anni. Si consiglia di spezzare la soluzione del problema in più funzioni.

Esercizio 4 **(4 punti)**

Una matrice NxM di interi (con N e M costanti predefinite) rappresenta la mappa di una nave da crociera. Ogni riga della matrice rappresenta un piano della nave e ogni casella una cabina del piano.

I numeri contenuti in ogni casella della matrice corrispondono alle persone che possono stare nella cabina corrispondente senza violare le norme di sicurezza. Le stanze marcate con 0 sono le “sale di salvataggio”, speciali cabine vuote che contengono i giubbotti di salvataggio della nave.

Si verifichi che su ogni piano ci sia un numero di “sale di salvataggio” tali per cui non debba succedere che più di 10 persone debbano usare la stessa “sala di salvataggio”. Si consiglia di spezzare la soluzione del problema in più funzioni.