Informatica A

**Cognome**

**Nome**

**Matricola o Codice studente**

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 ricalcare al momento della consegna).
4. È **vietato** utilizzare **calcolatrici** o **telefoni**. 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: 3 ore

**Esercizio 1 ( 2 punti )**

**Esercizio 2 ( 4 punti )**

**Esercizio 3 ( 4 punti )**

**Esercizio 4 ( 6 punti )**

**Esercizio 5 ( 12 punti )**

**Esercizio 6 ( 4 punti )**

Esercizio 1 ( 2 punti )

Si costruisca la tabella di verità della seguente espressione booleana.

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

Si rappresentino in complemento a due i numeri A = -55 e B = 76, se ne calcoli la differenza (A-B) e si indichi se si genera riporto sulla colonna dei bit più significativi e se si verifica overflow.

Esercizio 2 ( 4 punti )

Il seguente schema descrive la base di dati di un autonoleggio.

AUTO ( Targa, Modello, Marca )

CLIENTE ( CodiceFiscale, Nome, Cognome, Indirizzo, Telefono, Convenzione )

NOLEGGIO (CodiceFiscale, Targa, Data\_inizio, Data\_fine, Costogg, giorni, km )

Determinare i clienti che hanno noleggiato almeno due modelli diversi della stessa marca.

Il seguente schema descrive la base di dati di un autonoleggio.

AUTO ( Targa, Modello, Marca )

CLIENTE ( CodiceFiscale, Nome, Cognome, Indirizzo, Telefono, Convenzione )

NOLEGGIO (CodiceFiscale, Targa, Data\_inizio, Data\_fine, Costogg, giorni, km )

Determinare, per ogni autovettura, il noleggio (codice fiscale, targa e data inizio), iniziato nel 2016, nel quale sono stati percorsi più chilometri.

Esercizio 3 ( 4 punti )

Si dica cosa stampa il seguente codice e si dica quale è il massimo numero di chiamate di f aperte contemporaneamente sullo stack.

#include<stdio.h>

int f(int a, int b) {

if(a<b)

return a;

return f(a-b,b);

}

int main() {

int a[4]={1,2,3,4};

int x=0, y=0, k;

while(y==0) {

x++;

y=x;

for (k=0;k<4;k++)

if (f(x,a[k])!=0)

y=0;

}

printf ("%d\n",x);

}

Esercizio 4 ( 6 punti )

Scrivere una funzione che riceve una stringa di caratteri e restituisce 1 se la stringa è costituita da sequenze alternate di lunghezza pari di “a” e di “b”.

Ad esempio la stringa “aaaabbaaaaaabbbbaa” e “aaaa” rispettano la regola, mentre “aabaabbb” e “aaaabbccaabbbb” non la rispettano.

Esercizio 5 ( 12 punti )

Si considerino le seguenti definizioni di liste:

typedef struct nodo { int numero;

struct nodo \*next; } Nodo;

typedef Nodo \* Lista;

struct nodoCompresso { int numero, quanti;

struct nodoCompresso \*next; } NodoCompresso;

typedef NodoCompresso \* ListaCompressa;

Si codifichi in C la seguente funzione:

ListaCompressa comprimi(Lista A);

che riceve in input una lista di interi A e restituisce una nuova lista B “compressa” secondo la seguente regola: per ogni elemento di A si memorizza in B l’elemento stesso e il numero di ripetizioni consecutive in A.

Ad esempio se A = (3, 3, 3, 3, 2, 2, 3, 5, 5, 5) allora la lista restituita B = ((3, 4), (2, 2), (3, 1), (5, 3)).

Si codifichi in C la seguente funzione:

ListaCompressa comprimiTantissimo(Lista A);

che riceve in input una lista di interi A e restituisce una nuova lista B “compressa tantissimo ordinatamente” secondo la seguente regola: per ogni elemento di A si memorizza in B l’elemento stesso e il numero totale di ripetizioni in A ordinate in base al valore dell’elemento stesso.

Ad esempio se A = (**3, 3, 3, 3**, 2, 2, **3**, 5, 5, 5) allora la lista restituita B = ((**2**, 2), (**3**, 5), (**5**, 3)).

Esercizio 6 ( 4 punti )

Si considerino le seguenti definizioni:

struct nodoCompresso { char lettera;

int quanti;

struct nodoCompresso \*next; } NodoCompresso;

typedef NodoCompresso \* ListaCompressa;

typedef struct ET { char parola[1000];

struct ET \* left, \* right; } treeNode;

typedef treeNode \* Tree;

Si codifichi in C la seguente funzione:

ListaCompressa comprimiTantissimo(Tree T);

che riceve in input un albero T e restituisce una nuova lista “compressa tantissimo ordinatamente” secondo la seguente regola: per ogni lettera presente in qualche parola di T si memorizza nella lista la lettera e il numero totale di ripetizioni in T ordinate in base al valore della lettera stessa.

Ad esempio se T contiene “albero”, “mamma” e “casa” allora la lista restituita conterrà

((**a**, 5) , (**b**, 1), (**c**, 1) , (**e**, 1), (**l**, 1) , (**m**, 3), (**o**, 1) , (**r**, 1), (**s**, 1)).