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.

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

Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = 56dec e B = –123dec, 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:

#include<stdio.h>

#define D 4

float A(float V[], int k){

int i;

float s=0.0;

for(i=0;i<D; i=i+k) {

V[i]=i;

s=s+V[i];

}

k++;

return s;

}

int main(){

float V[]={0, 4.5, 0, 6};

int i, j=2;

char inutile;

for (i=1; i<D; i=i+1)

V[i]=V[i-1];

printf("%f\n", A(V,j));

for(i=0; i<D; i=i+1)

printf("%f ", V[i]);

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

fflush(stdin);

scanf("%c",&inutile);

return 0;

}

Esercizio 3 (4 punti)

Si considerino le seguenti dichiarazioni di tipi che definiscono le strutture dati per gestire una compagnia telefonica.

typedef struct {

int giorno, mese, anno;

} data;

typedef struct {

int ora, minuto, secondo;

} ora;

typedef struct {

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

char cittaDiResidenza[100];

data DataNascita;

} cliente;

typedef struct {

cliente cli;

data d;

ora o;

int durata; //in secondi

} telefonata;

typedef struct {

telefonata telefonate[1000];

int numTelefonate; //dice quante caselle dell’array contengono dati validi

} listaTelefonate;

Si definisca una funzione di prototipo

int f(listaTelefonate T);

che restituisce la somma della durata delle telefonate effettuate dal cliente che ha effettuato il minor numero di telefonate (per semplicità si supponga sia unico).

**Esercizio** 4 **(4 punti)**

Una matrice si dice “a V” se esiste una riga tale per cui partendo da essa e muovendosi sia verso l’alto sia verso il basso è sempre più alta la somma degli elementi delle righe che si incontrano.

Scrivere una funzione

int f(int M[][N])

che riceve in input una matrice di dimensione NxN (con N costante definita globalmente con una define) e restituisce 1 se la matrice è “a V”, 0 altrimenti.