



**Programmazione – CdS in Matematica
Appello d'esame 24 Settembre 2008**

Nome
Cognome
Matricola

INDICARE SUBITO NOME, COGNOME, E MATRICOLA

**NON è permesso (pena espulsione) usare la calcolatrice e consultare appunti e libri.
Scrivere le risposte e commentare i programmi CHIARAMENTE (la chiarezza sarà un
criterio determinante nella valutazione degli esercizi).**

----- PARTE 1 -----

Esercizio 1.1

Scrivere l'output del seguente programma C. (Indicare con '_' underscore gli eventuali spazi e con '<n>' la riga completamente vuota)

```
#include<stdio.h>

int main() {

    int a=b=c=0;
    c=++a;
    b=a++;
    if (b>a) printf("SI\n");
    else printf("NO\n");

    a=(b+=1);
    printf("%d %d\n", a, b);

    float x = 4.0;
    while (1) {
        printf("%f ", x);
        x /= 2;
        if (x == 1.0) break;
    }
    printf("\n");

    char str[15] = "ultimo-appello";
    int i;
    for (i=10; i>=0; i--)
        printf("%c ", str[i--]);
    printf("\n");
}
```

Esercizio 1.2

Implementare la funzione

```
float Determinante(float a[2][2]);
```

che, data in ingresso una matrice 2x2, ne restituisce il determinante.

Esercizio 1.3

Discutere le differenze e similitudini tra array e stringhe. Fare esempi chiarificatori.

Esercizio 1.4

Scrivere una funzione con prototipo

```
float StampaPi(int n);
```

che calcoli e stampi il valore di π usando la formula approssimata $\pi = \sum_{i=0}^n \frac{4 \cdot (-1)^i}{2i+1}$

Esercizio 1.5

Identificare e (se possibile) correggere gli errori in ognuna delle seguenti istruzioni.

```
1. printf("d d d", 1, 2, 3);
2. int m == (a>b)?a.b;
3. int x; printf(x);
4. int *p=3100;
5. int v[3]; for (int i=0; i<=3; i++) printf("%d ", v[i])
```

Esercizio 1.6

Descrivere le funzionalità (cosa calcola) e discutere le condizioni che devono essere verificate sui parametri delle seguenti funzioni. Esemplicare dettagliatamente la loro invocazione.

A)

```
float AAA(int n) {
    if (n==1) return 1;
    int i;
    for (i=n-1; i>1; i--)
        if (n%i==0) return 0;
    return 1;
}
```

B)

```
int BBB(int a[], int sx, int dx, int el) {
    int x;
    if (dx < sx) return -1;
    x = (dx + sx)/2;
    if (el < a[x]) return BBB(a, sx, x-1, el);
    else if (el == a[x]) return x;
    else return BBB(a, x+1, dx, el);
}
```

----- PARTE 2 -----

Esercizio 2.1

Si definisca una struttura dati (usando `struct`) per contenere le informazioni di un CD musicale. Si definisca un programma che dato un catalogo di CD (per esempio un array) stampi i dati dei CD corrispondenti ad un certo autore.

Esercizio 2.2

Sia data la seguente definizione di lista si implementino con tale struttura dati le operazioni standard di una coda non ancora definite (la realizzazione puo' essere sia ricorsiva che iterativa).

```
#include<stdio.h>

typedef struct elemento {
    float valore;
    struct elemento *next;
} Elemento;

typedef Elemento* List;

void Push(List *lista, Elemento e) {
    // aggiunge un elemento in fondo alla lista
}

void Pop(List *lista) {
    // rimuove l'elemento in testa alla lista
}

float Read(List lista) {
    // legge il primo elemento della lista
}

int Count(List lista) {
    // conta gli elementi della lista
}

void Incrementa(List lista, float f) {
    // incrementa tutti gli elementi della lista di un valore f
}
```

Esercizio 2.3

Si dia una descrizione sommaria (ma chiara) degli algoritmi di ordinamento conosciuti con riferimento anche alla loro complessita' computazionale.

Esercizio 2.4

Si dia un algoritmo ricorsivo per il calcolo del determinante di una matrice. Si ricorda che data una matrice di ordine N, il determinante si puo' calcolare con la formula di LaPlace,

$$Det(M) = \sum_{j=1}^N (-1)^{(i+j)} m_{ij} Det(M_{ij}), \text{ e } i \text{ e' una riga fissata.}$$

(n.b. M_{ij} e' la sottomatrice che si ottiene da M togliendo la riga i -esima e la colonna j -esima).