



**Programmazione – CdS in Matematica  
Appello d'esame 4 Luglio 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++;
    printf("%d %d %d\n", a, b, ++c);

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

    double x1 = 1.0, x2 = 10.0;
    for ( ; x1 < x2; x1 *= 2)
        printf("%d/%d", x1,x2);
    printf("\n");

    char str[21] = "Voglio passare questo esame\0 con 28\0 o 30";
    printf("lungh: %d\n", strlen(str));

    int i;
    for (i=0; i<21; i+=2) {
        printf("%c",str[i])
        printf("\n");
    }
}
```

## Esercizio 1.2

Descrivere (anche graficamente) passo x passo l'effetto del seguente frammento di programma.

```
int n=2;
float *p;
float *pluto = malloc(n*sizeof(float));
p = pluto++;
```

## Esercizio 1.3

Descrivere bene le caratteristiche del passaggio dei valori alle funzioni per indirizzo e per valore.

## Esercizio 1.4

Scrivere una funzione con prototipo

```
void StampaMatrice(float A[N][M]);
```

che stampi la matrice A e la sua trasposta

(n.b. N e M sono state precedentemente definite con una #define).

## Esercizio 1.5

Identificare e correggere gli errori in ognuna delle seguenti istruzioni.

1. `scanf("d",value);`
2. `first+second=third`
3. `if (x=y) printf("%d e' uguale a %d\n, x, y);`
4. `p+1=3;`
5. `int v[2]; for (int i=0; i<=2; 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. Esempificare dettagliatamente la loro invocazione.

**A)**

```
float AA(float *a, int n) {
    int i;
    float m=0,v=0;
    for (i=0; i<n; i++)
        m += a[i];
    m /= n;
    for (i=0; i<n; i++)
        v += (a[i]-m)*(a[i]-m);
    return v;
}
```

**B)**

```
int BB(int a) {
    if (a<10) return (a+1)%10;
    return 10*BB(a/10)+(a+1)%10;
}
```

**C)**

```
int CC(int m[N][N]) {
    int q=0, i, j;
    for (i=0; i<N; i++)
        for (j=0; j<N; j++)
            q = m[i][j] * m[j][i];
    return q;
}
```

**D)**

```
float DD(float *k, int n) {
    if (n==1) return k[0];
    float m = DD(k+1,n-1);
    return (k[0]<m)?k[0]:m;
}
```

## ----- PARTE 2 -----

### Esercizio 2.1

Dare un programma

```
void CalcolaSerie(double q, int n);
```

il più efficiente possibile (rispetto al numero di operazioni effettuate) per verificare che il valore di convergenza della serie:

$$S_n = \sum_{i=0}^n q^i$$

sia uguale a  $\frac{1}{1-q}$  se  $|q| < 1$ .

### Esercizio 2.2

Sia data la seguente definizione di lista si implementino con tale struttura dati le operazioni standard di uno stack 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 testa 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 di un valore f
}
```

### **Esercizio 2.3**

Si diano le descrizioni dettagliate degli algoritmi di ordinamento BUBBLESORT e MERGESORT e discuterne le differenze.

### **Esercizio 2.4**

Si definisca una matrice quadrata  $N \times N$  come array di puntatori. Si implementino poi 3 procedure non banali relative al calcolo matriciale (es. moltiplicazione di matrici, elevamento a potenza, trasposta). Si dichiarino esplicitamente la funzionalità calcolata e si commentino bene il codice.