



**Programmazione 1 – CdS in Matematica  
Appello d'esame 27 Marzo 2007**

Nome .....  
Cognome .....  
Matricola .....

**INDICARE NOME, COGNOME, E MATRICOLA SU OGNI FOGLIO UTILIZZATO**

**NON è permesso 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 vuota)

```
#include<stdio.h>

int main() {

    int a=13;

    int b=2;
    printf("%d\n",a/b);

    float d=2;
    printf("%f\n",a/d);

    char c;
    for (c='a'; c<='e'; c++)
        printf("%c",c);
    printf("\n\n");

    char str[20] = "ciao gente\0come va?\0";
    printf("lunghezza: %d\n", strlen(str));

    int i;
    for (i=0; i<20; i+=3)
        printf("%c",str[i]);
    printf("\n");

}
```

## Esercizio 1.2

Realizzare un programma che richiede un intero dispari (l) e disegna un quadrato di lato l, a righe intermitteni, utilizzando il carattere \* (asterisco).

Esempio con l=5:

```
*****
*   *
*****
*   *
*****
```

## Esercizio 1.3

Descrivere la sintassi e l'utilizzo del costrutto IF ELSE

## Esercizio 1.4

Descrivere le funzionalità e discutere le assunzioni che devono essere verificate sui parametri delle seguenti funzioni:

**A)**

```
char Gamma(char un_carattere, int key) {
    int a = (int) 'a';
    return (char) (a + ((int) un_carattere - a + key)%26);
}
```

**B)**

```
int Delta(char* pc) {
    int c=0;
    while (*pc) {c++; pc++;}
    return c;
}
```

**C)**

```
void Theta(char* pc) {
    if (!*pc) return;
    pc[0]=(*pc)++;
    Theta(pc+1);
}
```

**D)**

```
int Rho(int[N][M] m) {
    int q=0, i, j;
    for (i=0; i<N; i++)
        for (j=0; j<M; j++)
            if (m[i][j]>0)
                q += m[i][j];
            else
                q -= m[i][j];
    return q;
}
```

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

### Esercizio 2.1

Descrivere le similitudini e le differenze delle funzioni seguenti:

```
int Inc1(int val) { return val++; }
int Inc2(int *val) { return ++(*val); }
int Inc3(int *val) { return (*val)++; }
```

### Esercizio 2.2

Si definiscano le funzioni mancanti nel seguente programma per il gioco della “briscola”:

```
#include<stdio.h>

typedef enum {cuori, picche, quadri, fiori} semi;

typedef struct {
    int valore;
    semi seme;
} carta;

void InizializzaMazzo(carta mazzo[40]) {
    // definisce le carte del mazzo da briscola
}

void MescolaMazzo(carta mazzo[40]) {
    // mescola le carte nel mazzo
}

void OrdinaMazzo(carta mazzo[40]) {
    // ordina le carte per valore
}

int main() {

    srand(time(NULL));

    carta mazzo[40];

    InizializzaMazzo(mazzo);

    MescolaMazzo(mazzo);

    OrdinaMazzo(mazzo);
}
```

## Esercizio 2.3

Sia data la seguente definizione di lista si implementino con tale struttura dati le operazioni standard di uno stack (la realizzazione può 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
}
```