



**Programmazione – CdS in Matematica  
Appello d'esame 17 Marzo 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=1;
    float b=2;

    printf("%d\n",a/b);
    printf("%f\n",a/b);

    char c1,c2;
    for (c1='a'; c1<='e'; c1++)
        for (c2='a'; c2<='e'; c2++)
            if (c1==c2) printf("%c",c1);
    printf("\n");

    char str[20] = "Tanti auguri";
    int l=strlen(str);
    str[4]=str[l];
    printf("%s\n",str);

    int p,q,r=0;
    p=++r+1;
    q=r++-1;
    printf("%d %d %d\n", p, q, r);

    if (p=0)
        printf("%s\n", "SI");
    else
        printf("%s\n", "NO");
}
```

## Esercizio 1.2

Implementare le seguenti funzioni:

1) void LeggiArray(int a[], int n)

che richiede all'utente i valori da immettere in un array di lunghezza n

2) void DecArray(int a[], int n)

che decrementa tutti valori di un array di lunghezza n

## Esercizio 1.3

Descrivere bene la sintassi e l'utilizzo dei costrutti WHILE e DO-WHILE

## Esercizio 1.4

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

```
char s[]="parole";
char *p=s;
char *q=++p;
p[2] = 'f';
```

## Esercizio 1.5

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

**A)**

```
void A(int a[], int n) {
    if (n>0){
        A(a+1,n-1);
        printf("%d ", *a);
    }
}
```

**B)**

```
void B(char s1[],char s2[]){
    int i,j;
    for (i=0; s1[i]!='\0'; i++);
    for (j=0; s2[j]!='\0'; j++)
        s1[i+j]=s2[j];
    s1[i+j]='\0';
}
```

**C)**

```
void D(int m[N][N]) {
    int i, j;
    for (i=0; i<N; i++)
        for (j=0; j<N; j++)
            m[i][j]=m[j][i];
}
```

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

### Esercizio 2.1

Si vuole mostrare empiricamente che la serie armonica *non e' convergente*, ovvero che per ogni valore reale  $B$ , esiste un  $n$  tale che

$$S_n = \sum_{i=1}^n \frac{1}{i} \geq B$$

A tale scopo, realizzare la funzione C

```
int NSA(double B);
```

che calcola il valore di  $n$  minimo necessario perche' la condizione sopra risulti vera.

### Esercizio 2.2

Sia data la seguente definizione di lista concatenata. 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* Stack;

void Push(Stack *s, float f) {
    // aggiunge un elemento con valore f in testa alla lista
}
void Pop(Stack *s) {
    // rimuove l'elemento in testa alla lista
}
float Read(Stack *s) {
    // ritorna il primo elemento della lista
}
int Count(Stack *s) {
    // conta gli elementi della lista
}
void Incrementa(Stack *s, float f) {
    // incrementa tutti gli elementi di un valore f
}
```

### Esercizio 2.3

Si dia un programma ricorsivo che effettui una "ricerca binaria" su di un vettore di interi di lunghezza  $n$  ordinato in modo crescente (ritornando il puntatore all'elemento, o il valore 0 se l'elemento non esiste). In particolare, si implementi la funzione

```
int* RicercaBin(int val, int vet[], int n);
```