

Laboratorio di Informatica

Università di Catania
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
Corso di Laurea in Ingegneria Informatica
N.O., A.A. 2002/03

Laboratorio di Informatica

Ing. Ferdinando Portuese

1^a Prova intermedia

Vengono proposti 3 gruppi di Domande:

- ✓ **Gruppo A** : domande di carattere Generale (verifica conoscenze acquisite) Max. 16 punti
- ✓ **Gruppo B** : domande di Analisi Max. 10 punti
- ✓ **Gruppo C** : domande di Sintesi Max. 4 punti

- Accanto ad ogni domanda è indicato il relativo punteggio massimo attribuibile.
- La sufficienza alla prova si ottiene se il punteggio totalizzato è almeno **18**.
- La sufficienza in questa prova è necessaria per l'ammissione alla seconda prova (pratica).
- Tra le risposte proposte per ciascuna domanda più di una potrebbe essere esatta.
- Utilizzate la riga "**Eventuale commento** _____" per motivare eventualmente la risposta.
- Il tempo a disposizione è di 2 ore.
- Non è concesso l'uso di testi o appunti.

Studente : _____ Matricola _____

Anno di corso : _____

Firma

Laboratorio di Informatica
Gruppo A

A1 – (0,5 punti) Supponiamo di dover assegnare due strutture identiche S1 ed S2 come sotto definite; quali delle seguenti istruzioni **non è consentita** per effettuare l'assegnamento ?

<pre>struct prova { int A; float B; } struct prova S1; struct prova S2;</pre>	<input type="checkbox"/> S1 = S2; <input type="checkbox"/> S1.A = S2.A; S1.B = S2.B; <input checked="" type="checkbox"/> S1 == S2; <i>Eventuale commento</i> _____
--	--

Eventuale commento == è un operatore di confronto e non di assegnamento _____

A2 – (0,5 punti) Trova gli errori contenuti nel seguente frammento di programma ed indicane la correzione :

<pre>struct Persona { char nome[25]; char cognome[25]; char matricola[8]; }; Dipendente Persona; Dipendente.nome = "Paolo";</pre>
--

Errore: **Dipendente Persona;** _____ **Correzione:** **struct Persona Dipendente** _____

A3 - (2,5 punti) Siano date le seguenti dichiarazioni di strutture e definizione di variabile:

```
struct canzone{
    char titolo[30];
    long int durata; /* espressa in secondi */
};
struct CD {
    char titolo[30];
    int numCanzoni;
    struct canzone canzoni[25];
    char autore[30];
    float prezzo;
};
```

```
struct CD miacollezione[30];
```

Scrivere le istruzioni per

- 1) Assegnare al terzo CD di miacollezione il prezzo di €13 (assegnare 13 al campo prezzo).
 ___ **miacollezione[2].prezzo = 13;** _____
- 2) Assegnare al campo numCanzoni del quinto CD di miacollezione il valore 15.
 ___ **miacollezione[4].numcanzoni = 15;** _____
- 3) Assegnare alla prima canzone del primo CD di miacollezione la durata di 2min e 30sec.
 ___ **miacollezione[0].canzoni[0].durata = 150;** _____
- 4) Assegnare al primo carattere del titolo della terza canzone del sestCD il valore 'L'.
 ___ **miacollezione[5].canzoni[2].titolo[0] = 'L';** _____
- 5) Assegnare al secondo carattere del titolo del quarto CD il valore 'd'.
 ___ **miacollezione[3].titolo[1] = 'd';** _____

Laboratorio di Informatica

A4 - (1 punto) Se s1 ed s2 sono due stringhe `char s1[25]; char s2[25];` e si scrive `s1 = s2;` cosa succede?

- Tutto il contenuto di s2 viene copiato in s1
 Il primo elemento di s2 viene copiato nel primo elemento di s1
 Si ottiene un errore di compilazione

Eventuale commento s1 (nome dell'array) contiene l'indirizzo del primo elemento ma è una costante

A5 - (0,5 punti) Sia dato il seguente frammento di programma:

```
int *fp();
```

- è sintatticamente scorretto
 è la dichiarazione di una funzione senza parametri che restituisce un puntatore a intero
 è la dichiarazione di un puntatore a funzione

A6 - (1 punto) Sia dato il seguente frammento di programma:

```
int x=6;
int y;
&y=++x;
```

- il frammento di programma contiene un errore sintattico**
 il frammento di programma contiene più di un errore sintattico
 il frammento di programma è sintatticamente corretto, e dopo la sua esecuzione x e y valgono 7

Eventuale commento l'indirizzo della variabile y non può stare nella parte sinistra di un'assegnazione (l'indirizzo di y è una costante)

A7 - (2 punti) Siano x e y interi e z reale. Siano inoltre p e q puntatori a interi ed r puntatore a reale. Indicare in tabella i cambiamenti operati da ciascuna istruzione. Nel caso in cui l'istruzione si ritiene errata sintatticamente, riportare nella colonna errore un *.

Istruzione	x	y	z	*p	*q	*r	errore
x = 10; y = 5;	10	5					
z = x;			10				
p = &y;				5			
q = p;					5		
r = q;							* (1)
*r = 12;						12	
z += *r;			22				
p = &(x+2);							* (2)
*q = *z + 1;							* (3)

Eventuale commento (1) r e q puntano ad oggetti di tipo diverso

(2) non ha senso l'indirizzo dell'espressione (x+2). & si applica a variabili

(3) non è lecito scrivere *z (l'operatore * si applica ad un puntatore e non ad una variabile reale)

Laboratorio di Informatica

A8 - (2,5 punti) Siano **X** e **Y** due vettori di interi. Siano inoltre **p** e **q** puntatori a interi.

Dopo aver segnato le istruzioni che si ritengono errate (con) seguire l'evoluzione delle istruzioni (ignorando quelle errate) e riportare i contenuti dei due vettori:

<pre>int n = 2; int X[5] = {6, 7, 8, 9, 10}; int Y[5] = {1, 2, 3, 4, 5};</pre>	<input type="checkbox"/> p = X ; <input type="checkbox"/> *(p+n) = Y[n] ; <input checked="" type="checkbox"/> Y = p ; <input type="checkbox"/> q = &Y[4]; <input type="checkbox"/> n = *q; <input type="checkbox"/> *(q-2) = n; <input type="checkbox"/> p = Y; <input type="checkbox"/> *p = n;												
<p>Valori a fine codice:</p> <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 5px;">X</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">6</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">7</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">3</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">9</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">10</td> </tr> <tr> <td style="text-align: center; padding: 5px;">Y</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">5</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">2</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">5</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">4</td> <td style="border: 1px solid black; padding: 2px 10px; text-align: center;">5</td> </tr> </table>	X	6	7	3	9	10	Y	5	2	5	4	5	
X	6	7	3	9	10								
Y	5	2	5	4	5								

A9 - (0,5 punti) Di che tipo è la variabile **ft** restituita dalla funzione `fopen` qui di seguito riportata (si scriva anche la dichiarazione):

```
ft = fopen("dati.txt", "r");
```

Tipo restituito: Puntatore a file _____

Dichiarazione: FILE *ft _____

A10 - (1 punto) Dato il file "dati.txt" di seguito riportato e contenente due righe di testo, se `ch` è di tipo carattere e `fp` è un puntatore a file, quante volte cicla il costrutto `do ... while` contenuto nel frammento di codice?

<pre>fp = fopen("dati.txt", "r"); do fscanf(fp, "%c", &ch); while (!feof(fp)); fclose(fp);</pre>	<p style="text-align: center;">dati.txt</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 5px;">pippo-</td> </tr> <tr> <td style="padding: 5px;">pluto-</td> </tr> </table>	pippo-	pluto-
pippo-			
pluto-			

Risposta: 15 = (12 caratteri + 2 EOL + 1 EOF) _____

A11 - (1 punto) Data la seguente funzione dove il passaggio di parametri avviene per valore, trasformarla in una funzione dove il passaggio è per indirizzo :

<pre>int Incremento(int a, int b) { int somma; a++; b++; somma = a + b; return somma; }</pre>	<pre>int Incremento(int *a, int *b) { int somma; *a = *a + 1; *b = *b + 1; somma = *a + *b; return somma; }</pre>
---	---

Eventuale commento *a = *a + 1 (*a)++ (*a) += 1 sono equivalenti

Laboratorio di Informatica

A12 – (1 punto) Sia dato il seguente programma:

```
#include<stdio.h>

void stampa(int b);
int f(int c);

main (){
    int a=3;
    stampa(a);
}

void stampa(int b)
{printf("%d",f(b));}

int f(int c){return (c*c);}
```

- il programma contiene un errore sintattico
 il programma contiene più di un errore sintattico
 l'esecuzione del programma provoca la stampa di 9

Eventuale commento _____

A13 – (1 punto) Sia dato il seguente programma:

```
#include <stdio.h>
#define N 10

main(){
    int k, v[N] = {0,1,2,3,4,5,6,7,8,9};
    for (k=1; k<=N; k++) printf("%d ",*(v+k));
    return;
}
```

esso è corretto sia dal punto di vista sintattico che logico?

- Sì
 No Perché ? Quando $k = N$ si tenta di accedere all'elemento $v[N]$ che non esiste (o meglio è casuale)

A14 - (1 punto) Sia dato il seguente frammento di programma:

```
if ( (char *) malloc(sizeof(int)) <= (char *)malloc(sizeof(long int)) )
    printf("%d\n", 0);
else printf("%d\n", 1);
```

- è sintatticamente scorretto
 è sintatticamente corretto ed il valore di output è sicuramente zero
 è sintatticamente corretto ed il valore di output è sicuramente uno
 è sintatticamente corretto ed il valore di output può essere sia zero che uno

Eventuale commento **Indipendentemente dalla dimensione di memoria allocata, i 2 elementi a confronto sono puntatori i cui valori sono assegnati dal gestore della memoria**

Laboratorio di Informatica

Gruppo B

B1 – (2,5 punti) Sia dato il seguente programma:

<pre>#include <stdio.h> #define D 4 float A(float *V, int k) { int i; float s=0.0; for (i=0;i<D; i+=k) { V[i] = i; s=s+V[i]; } return s; }</pre>	<pre>void main() { float V[] = {1.1, 2.2, 3.3, 4.4}; int i; for (i=1; i<D; i++) V[i] = V[i-1]; printf("%f\n", A(V,2)); for (i=0; i<D; i++) printf("%f\n", V[i]); return; }</pre>
---	--

Qual è l'uscita del programma? =>

2.0
0.0 1.1 2.0 1.1

B2 - (2,5 punti) Dato il seguente programma, indicare quanto qui sotto richiesto:

```
#include <stdio.h>
#define N 6
void main()
{
    int V[] = {0, 0, 0, 0, 0, 0};
    int i;
    for (i=0; i<N; i += 2)
        V[i] = V[i] + 3*i;
    for (i=0; i<N; i++)
        if (V[i]) printf("%d\t", V[i]);
        else printf("%d\t", i-V[i]);
    return;
}
```

Valori di V dopo il 1° ciclo:

0	0	6	0	12	0
----------	----------	----------	----------	-----------	----------

Output:

0 1 6 3 12 5

Laboratorio di Informatica

B3 - (2,5 punti) Nell'ipotesi che n sia il numero degli elementi contenuti nel vettore vet e che i sia un valore compreso tra 0 e n , dire qual è l'output del seguente frammento di programma:

```
void mystery(int *vet, int n, int i)
{
    int j, b;
    b = vet[i];
    for(j=i+1; j<n; j++)
        vet[j-1] = vet[j];
    vet[n-1] = b;
}
```

```
.....
int V[6] = {10, 20, 30, 40, 50, 60};
.....
mystery(V, 6, 2);
for i=0; i<6;i++) printf(("%d ",V[i]);
```

Output:

10	20	40	50	60	30
----	----	----	----	----	----

B4 - (2,5 punti) Si indichino, nel giusto ordine, i valori stampati dalle istruzioni (1) e (2) del seguente programma C, motivando la risposta data.

```
#include <stdio.h>
#define L 4
#define C 3
int N=L*C;
void main () {
    int MAT[L][C]={ {2,3,4}, {3,4,5}, {4,5,6}, {5,6,7} };
    void Fun(int *v, int dim);
    int i, j;
    Fun(MAT,L*C);
    printf("%d\n",N); /* (1) */
    for (i = 0; i < L; i++)
    { for (j = 0; j < C; j++)
        printf(" %d\t",MAT[i][j]); /* (2) */
        printf("\n"); }
}
```

```
void Fun(int *p, int size) {
    int i;
    N=N+1;
    for (i=0; i<size; i=i+3)
        *(p+i)= *(p+i)*10;
    return;
}
```

(1) :

13 _____

(2) :

20	3	4
30	4	5
40	5	6
50	6	7

Eventuale commento _____

Laboratorio di Informatica

Gruppo C

C1 - (4 punti) Scrivere una funzione che, dato in ingresso un array **V** di **N** numeri reali, ritorni lo stesso vettore **V** contenente gli inversi dei numeri presenti nell'array.

Gli zeri devono essere ignorati.

Ad esempio se **V** contiene i valori

{ 10 , 0 , 5 , 2 }

la funzione dovrà ritornare

{ 0.1 0.2 0.5 }

Soluzione:

```
void Inversi(float *V, int *N) {
    int i, k, N1 = *N;
    for (i=0; i<*N; i++)                /* calcola gli inversi */
        if (V[i]!=0.0) V[i]= 1.0/(V[i]);
        else N1--;                       /* N1 sarà la nuova dimensione di V */
    i = 0;
    while (i<N1) {
        if (V[i]==0.0) {
            k = i;
            while (k<*N-1) V[k++]=V[k+1]; /* shifta gli elementi indietro */
            V[*N-1] = 0.0;
        }
        i++;
    }
    *N = N1;                             /* aggiorna la dimensione di V */
}
```

Firma
