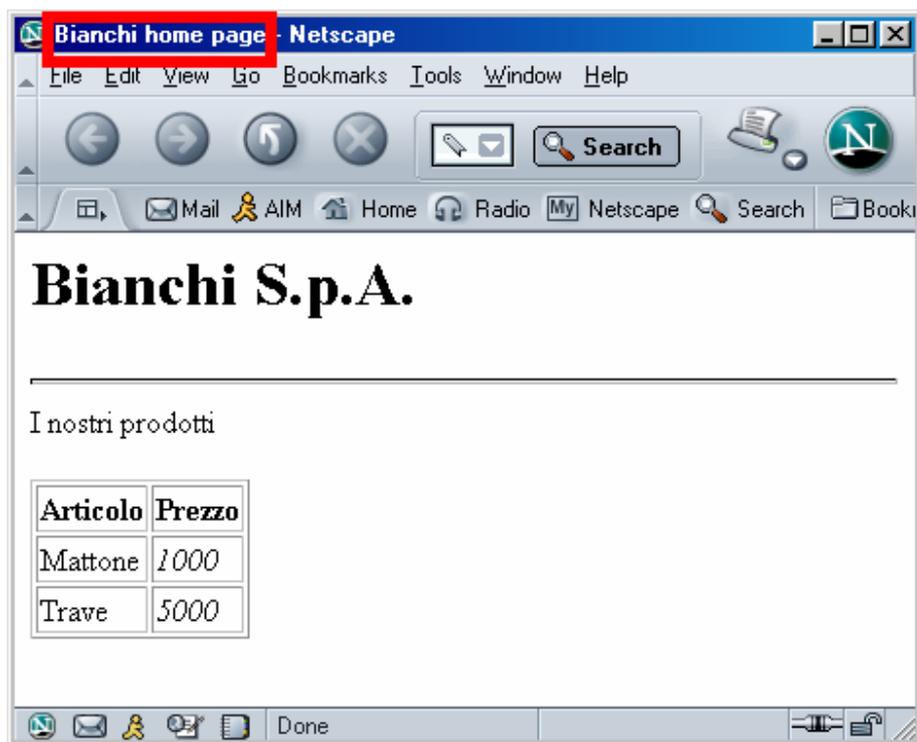


COGNOME e NOME

Matricola

1. **HTML** - Scrivere il codice HTML che produce la pagina sotto raffigurata. Nella barra del titolo del browser compare la dicitura “Bianchi home page” (vedi riquadro in figura). Nella pagina, il titolo “Bianchi S.p.A.” è di livello 1. Vi è poi una riga orizzontale. La prima riga della tabella adotta il formato di intestazione. I prezzi che compaiono in tabella sono in corsivo. **(punti 2)**



```
<html>
<head>
  <title>Bianchi home page</title>
</head>
<body>
<h1>Bianchi S.p.A.</h1>
<hr>I nostri prodotti<br>
<br>
<table border="1">
  <tr>
    <th>Articolo</th>
    <th>Prezzo</th>
  </tr>
  <tr>
    <td>Mattone</td>
    <td><i>1000</i></td>
  </tr>
  <tr>
    <td>Trave</td>
    <td><i>5000</i></td>
  </tr>
</table>
</body>
</html>
```

**Politecnico di Milano - Anno Accademico 2003-2004 - Informatica C**  
**Prima prova in itinere – 6 Maggio 2004 – Elaborato A**

|                |  |
|----------------|--|
| COGNOME e NOME |  |
| Matricola      |  |

2. **Costrutti condizionali** –Trasformare il seguente costrutto if in un equivalente costrutto switch. **(punti 1)**

|  |  |
|--|--|
| <pre>if(a==8)printf("gennaio\n"); else if(a==2)printf("febbraio\n"); else printf("mese sconosciuto!\n");</pre> | <pre>switch(a){ case 8: printf("gennaio\n"); break; case 2: printf("febbraio\n"); break; default: printf("mese sconosciuto!\n"); }</pre> |
|--|--|

3. **Operatori bitwise** - Che valore stampa questo spezzone di programma? **(punti 1) R: 0**

```
int i=10;
i=i<<2;
i=i&2;
printf("%d\n",i);
```

\* \* \*

Il seguente programma dovrebbe ricevere da tastiera un minimo e un massimo e poi stampare tutti i numeri interi compresi tra il minimo e il massimo, estremi compresi, andando a capo dopo ogni numero. Quando da tastiera viene inserito un minimo maggiore del massimo, il programma rifiuta l'immissione come non valida e questa viene ripetuta fino a quando i valori inseriti non sono accettabili. Il programma contiene però degli errori sintattici e/o logici. Esaminarlo attentamente e rispondere alle domande sotto riportate.

```
#include <stdio.h>;           (A)<-- punto e virgola

int main                       (B)<-- mancano parentesi (tonde)
{
    int min;
    int max;                   (C)<-- manca dichiarazione variabile 'i'
    do{
        printf("Valore iniziale:");
        scanf("%d",&min);
        printf("Valore finale:");
        scanf("%d",&max);
    }while(min>max);
    for(i=min; i<max; i++);<- 1.SBAGLIATA CONDIZIONE DI CICLO '<'
        printf("%d\n",i);      2.SBAGLIATO IL PUNTO E VIRGOLA
    return 0;
}
```

**Politecnico di Milano - Anno Accademico 2003-2004 - Informatica C**  
**Prima prova in itinere – 6 Maggio 2004 – Elaborato A**

|                |  |
|----------------|--|
| COGNOME e NOME |  |
| Matricola      |  |

4. Indicare sul listato (con la sigla “SINT”) e correggere gli errori che sarebbero segnalati dal compilatore compilando il programma.(punti 2) **R: v. note in rosso (3 errori da A a C)**
5. Dire se, una volta corretti gli errori di sintassi (individuati al punto precedente) e compilato il programma, il funzionamento sarebbe effettivamente quello descritto. In caso contrario, indicare e correggere sul listato (con la sigla “LOG”) gli errori NON segnalati immediatamente dal compilatore, ma evidenti solo in fase di esecuzione, a causa dei quali il programma si comporterebbe in modo scorretto o comunque diverso da quello descritto.(punti 2) **R: vedi note in rosso (2 errori logici da 1 a 2)**
6. Scrivere un frammento di programma che:
- dichiara una variabile di tipo idoneo a contenere valori come 3.25 o 0.01 senza perdita d’informazione;
  - legge da tastiera un valore per tale variabile;
  - stampa il valore della variabile in modo tale da visualizzare in ogni caso 3 cifre decimali e di occupare complessivamente un minimo di 7 caratteri (contando le cifre prima e dopo il punto decimale e il punto decimale stesso). (punti 1)

```
float a;
scanf("%f", &a);
printf("%7.3f", a);
```

**7. Manipolazione di stringhe**

Esaminare la seguente sequenza di istruzioni. Si supponga che faccia parte di un programma e **che venga eseguita nell’ordine dato qui sotto.**

Con riferimento a ogni istruzione, rispondere alle domande negli spazi alla destra di ogni riga del codice. (Punti 2)

| <i>istruzione</i>                                       | <i>quesito</i>                                       | <i>risposta:</i> |
|---|--|------------------|
| char a[20]; char b[20]; int x;                          | ----   | ----             |
| strcpy(b, "xyz");                                       | che valore contiene <b>b</b> dopo questa istruzione? | "xyz"            |
| strcpy(a, "abcd");                                      | che valore contiene <b>a</b> dopo questa istruzione? | "abcd"           |
| strcat(a,b);  | che valore contiene <b>a</b> dopo questa istruzione? | "abcdxyz"        |
| x=strlen(a);  | che valore assume <b>x</b> dopo questa istruzione?   | 7                |
| if(strstr(a, "yz")==0)printf("1");<br>else printf("2"); | che cosa viene visualizzato su schermo?              | 2                |

**Politecnico di Milano - Anno Accademico 2003-2004 - Informatica C**  
**Prima prova in itinere – 6 Maggio 2004 – Elaborato A**

|                |  |
|----------------|--|
| COGNOME e NOME |  |
| Matricola      |  |

8. Completare il seguente programma in modo tale che la sua esecuzione produca questo output:  
**3 4 5 7 8 (punti 1)**

```
int i;
for( i=3 ; i<=8 ; i++ )
{
    if( i!=6 ) printf("%d ",i);
}
```

**9. Strutture dati: struct e union. (punti 2)**

Un archivio aziendale costituito da un array di 100 elementi di tipo `elemento_archivio` è destinato a contenere informazioni su due diversi tipi di entità: persone o attrezzature.

Per le persone vengono memorizzati il nominativo (lungo al massimo 20 caratteri) e il codice fiscale. Per le attrezzature vengono memorizzati una descrizione testuale (massimo 30 caratteri) e un valore di inventario che è sempre compreso tra 0 e 60.000 euro ed è sempre arrotondato a un importo privo di decimali.

Definire il tipo di dato `elemento_archivio` utilizzando opportunamente struct e/o union. Fare in modo di sprecare meno memoria possibile.

```
struct dati_persona_s
{
    char nome[21]; /* 20 + \0 */
    char codfisc[17]; /* 16 + \0 */
};

struct dati_attrezzatura_s
{
    char descr [31]; /* 30 + \0 */
    unsigned int valore; /* 16 bit unsigned=max 65535 */
};

union dati_u
{
    struct dati_persona_s persona;
    struct dati_attrezzatura_s attrezzatura;
};

typedef struct elemento_archivio_s
{
    char tipo; /* 'p' per persona, 'a' per attrezzatura */
    union dati_u dati;
} elemento_archivio;
```