

**Politecnico di Milano - Anno Accademico 2003-2004 - Informatica C**  
**Appello 17 Settembre 2004 – REC2**

COGNOME e NOME	
Matricola	

### 1. Algoritmi (punti 3)

Scrivere una funzione `contiene_doppioni(char *a)` che stabilisce se la stringa passata come parametro contiene almeno un carattere che ricorre due o più volte nella stringa. Per esempio, CASA contiene un carattere ripetuto ("A"), quindi la funzione deve ritornare 1 (vero). ASTRATTO contiene due lettere (A e T) ripetute (rispettivamente due e tre volte); anche in questo caso la funzione dovrà quindi ritornare 1 (non importa quante sono le lettere ripetute nè quante volte si ripetono). Invece, per la parole LAGO e DOMANI, la funzione deve ritornare 0 (falso) perchè ogni carattere compare una sola volta.

*Suggerimento: per ogni carattere della stringa, si cerchi se nella parte successiva della stringa quel carattere è presente. Se viene trovato, ritornare 1. Altrimenti riprovare con il carattere seguente e ripetere l'analisi sulla parte successiva della stringa. Alla fine, se non è stata trovata alcuna ripetizione, ritornare 0.*

*Per controllare se un carattere è contenuto in una stringa si usi la funzione `strchr(char *stringa, char carattere)` che ritorna NULL se il carattere NON viene trovato nella stringa. **Punti 3** per una implementazione corretta di tipo iterativo. **BONUS 1** punto se viene invece proposta l'implementazione ricorsiva. **BONUS 3** punti se vengono proposte entrambe le implementazioni.*

Implementazione iterativa

```
int contiene_doppioni_iterativa(char * a)
{
    int i,j;
    if(strlen(a)<=1)return 0; /* se e' lunga 0 o 1 non puo' contenere doppioni..*/
    for(i=0; i<strlen(a)-1; i++)
        if(strchr(a+i+1,a[i])!=0) return 1;
    return 0;
}
```

Implementazione ricorsiva

```
int contiene_doppioni_ricorsiva(char * a)
{
    if(strlen(a)<=0)return 0; /* se e' lunga 0 o 1 non puo' contenere doppioni..*/
    if(strchr(a+1,a[0])!=0) return 1; /* trovato un doppione */
    else return contiene_doppioni_ricorsiva(a+1); /* prova con prossimo car. */
}
```

COGNOME e NOME	
Matricola	

## 2. If e Switch (punti 1+2)

Scrivere un costrutto if e un costrutto switch equivalente che producano il seguente effetto:

Se a vale 3 oppure 5, stampare il doppio di a.

Se a vale 11, stampare il quadrato di a.

Per gli altri valori di a, stampare "CIAO" se a è pari.

### Costrutto if

```
if((a==3) || (a==5))
    printf("%d", a*2);
else if(a==11)
    printf("%d", a*a);
else if(a%2 == 0)
    printf("CIAO");
```

### Costrutto switch (VIETATO USARE IF)

```
switch(a)
{
    case 3:
    case 5:
        printf("%d", a*2);
        break;
    case 11:
        printf("%d", a*a);
        break;
    default:
        switch(a%2)
        {
            case 0:
                printf("CIAO"); break;
        }
}
```

## 3. File (punti 2)

Si scriva un frammento di codice C che cerca di aprire in scrittura il file "dati.txt", di scrivervi una stringa e infine di chiuderlo. Si stampi un messaggio di errore nel caso in cui l'apertura fallisca.

```
FILE * f;
f=fopen("dati.txt", "w+");
if(f==NULL)
{
    printf("Impossibile aprire in scrittura il file");
    return;
}
fprintf(f, "%s", "stringa qualsiasi");
fclose(f);
```

## 4. Puntatori e strutture dati

Scrivere una dichiarazione di un array **dati** di 10 stringhe. La lunghezza massima di ogni stringa è di 12 caratteri.

```
char dati[10][13];
```

Scrivere una istruzione che stampa il terzo carattere della sesta stringa di tale array.

```
printf("%c", dati[5][2]);
```

Scrivere una istruzione che stampa "UGUALI" se la seconda e la ottava stringa sono uguali.

```
if(!(strcmp(dati[1], dati[7])))
    printf("UGUALI");
```

Scrivere una istruzione che assegna la stringa "PROVA" alla quinta stringa dell'array.

```
strcpy(dati[4], "PROVA")
```