

Esercizio 1

Elencare i diversi casi di visibilita' delle variabili in un programma C. Dettagliare i vari casi con esempi di frammenti di codice.

Esercizio 2

Descrivere il ruolo del sistema operativo nei calcolatori, descrivendo i diversi strati software usualmente presenti su di un calcolatore.

Esercizio 3

Leggere attentamente il seguente listato di codice. Il main presenta svariati errori; dopo averli individuati e motivati, correggerli riscrivendo la funzione main e illustrare cosa stampa il programma dopo aver letto la stringa "InformaticaGenerale".

```
1 #include <stdio.h>
2
3 int fun(char *s)
4 {
5     int n, i=0, j=0;
6     char str[20];
7     for(n=0; s[n]!='\0'; n++);
8     for(i=n; i>n/2; i--) {
9         s[n-i-1]=s[i];
10    }
11    s[i]='\0';
12    return i;
13 }
14 int main(void)
15 {
16     scanf("%20s", str[]);
17     fun(str);
18     printf("Ecco la stringa ottenuta ed il valore:", str, fun(str));
19     return(0);
20 }
```

Correzione

Anzitutto la variabile **str** non e' dichiarata. Si poteva dichiararla come variabile globale oppure nel main. Il fatto che sia dichiarata anche in fun non implica nulla. La dichiariamo di 21 visto che nella scanf leggo 20 caratteri (terminatore escluso, o la dichiaro di 20 ma cambia scanf leggendo max 19 caratteri).

Nella scanf **str[]** e' un errore. Occorre passare il puntatore dell'inizio stringa quindi solo **str**.

Nella printf devo mettere un template valido.

Inoltre la funzione fun va richiamata una volta sola (per ottenere la soluzione sotto).

Il main corretto e' il seguente e dopo aver letto la stringa "InformaticaGenerale" stampa
La stringa modificata e' elareneGa ed e' lunga 9.

Chi ha segnalato l'errore di segmentation fault dovuto a **s[-1]** ha sbagliato (l'errore viene segnalato se sfioriamo in eccesso la memoria assegnata, ma nella memoria antecedente le variabili c'e' codice del programma, o altre variabili, quindi e' memoria assegnata al ns processo, non ad altri) ma se correttamente motivato e' stato valutato sufficiente.

```
int main(void)
{
    char str[21];
    int val = fun(str);
    scanf("%20[A-Za-z ]", str);
    printf("Ecco la stringa ed il valore %s, %d.\n", str, val);
    return(0);
}
```