Esercizio 1

Quale utilita' hanno i tipi di dati in un linguaggio di programmazione? (si discuta il caso concreto del linguaggio C).

Correzione

La tipizzazione comporta diversi vantaggi, in breve: ottimizzazione della memoria (utilizzo di un short unsigned se il numero che ci si aspetta e' un contatore "piccolo"), controllo della semantica (eventuali warning di operazioni di casting segnalate in fase di compilazione), maggior precisione e affibilita' (se necessarie), meccanismo dei puntatori legato all'algebra dei puntatori.

Esercizio 2

1) Dato il seguente codice

```
...
   float a;
   double b;

   a = 1.0 / 5;
   b = 0.2;
...
```

spiegare come mai il test `if (a == b)` da risultato falso.

Correzione

Si noti che float e double non rappresentano numeri reali, ma una loro approssimazione... E l'approssimazione rappresentata da double e' diversa da quella rappresentata da float, quindi il test fallisce sempre (se non fallisce, c'e' un errore nel compilatore!).

Se le variabili fossero state tutte e due float (o tutte e due double) o se il testo avesse detto `1.0 / 2` e `0.5`, allora il test avrebbe sempre dato risultato vero.

Infatti con una semplice stampa `printf("a=%.9f b=%.9lf\n",a,b);` otteniamo `a=0.200000003 b=0.200000000` quindi i due valori differiscono, seppur in minima parte.

NON e' vero che il test fallisce perche' la divisione e' tra interi, infatti essendo tra un float e un interno il risultato sara' float! La variabile `a` varrebbe 0 solo se avessimo scritto `a=1/5`.

Esercizio 3

Il candidato analizzi il listato di codice sorgente C e risponda, motivando ogni risposta, alle seguenti domande in calce.

```
1  #include <stdio.h>
2
3  int fun(char *a)
4  {
5    int n=0;
6    for(i=1;a[i]!='\0';i++)
7      n=2*n+a[i]-'0';
8    return n;
9  }
10
11 int main(void)
12 {
13     int k;
14     char str[20];
15     scanf("%[01]",str);
16     k=fun(str);
17     printf("%d %o\n",k,k);
18     exit(0);
19 }
```

- La funzione `fun` e il `main` sono sintatticamente corretti? Se no, indicare gli errori individuati e le correzioni del caso.
- A cosa corrisponde il valore ritornato dalla funzione `fun`?
- Cosa stampa la funzione main dopo aver letto la stringa `"101101"`?

Correzione

- La funzione `fun` non e' sintatticamente corretta. Manca la dichiarazione di `i`. NON e' un errore il `-'0'`, anzi, e' fondamentale per calcolare il valore di ritorno esatto in modo da contare i caratteri '0' come 0 e '1' come 1 (invece che con il loro valore ASCII). Si poteva commentare che per usare `exit` occorre includere la libreria `stdlib.h`.
- Il valore ritornato dalla funzione `fun` corrisponde al valore in base decimale del valore espresso in base binaria passato come parametro, se il ciclo veniva corretto inizializzando `i` a 0. Similmente, qualora `i` fosse partito da 1 veniva comunque calcolato il corrispondente binario ignorando il bit piu' significativo (come se fosse un bit di segno e ritornando il valore assoluto del numero corrispondente).
- La funzione main dopo aver letto la stringa `"101101"` stampa il corrispondere decimale e ottale, cioe' `45 55` nel caso in cui `i` parta da 0, oppure `13 15` nel caso in cui `i` parta da 1.