

```
/*
 * Corso di Fondamenti di Informatica
 * Esercizio sull'uso di argv e argc
 */
#include <stdio.h> /* per poter usare la funzione strlen */
/*
Note sull'intestazione della funzione main:
1. una intestazione alternativa, del tutto equivalente a
   Ã..
   int main(int argc, char **argv)
2. i nomi (argv e argc) utilizzati per i parametri della funzione main
   fanno parte dello standard C. Il programma funzionerebbe ugualmente con
   nomi diversi, ma occorre attenersi allo standard.
*/
int main (int argc, char *argv[])
{
  int ContaArgomenti;
  /* usato per scegliere tra gli argomenti passati al programma tramite il
   * comando usato per eseguirlo */
  float ArgomentoFloat;
  /* risultato del tentativo di estrarre un float da un argomento */
  int Successo;
  /* vale 1 se e' stata compiuta con successo l'estrazione di un float da
   * una stringa, 0 altrimenti */
  printf("\nGli elementi del comando appena inserito si trovano nell'array argv[.].");
  printf ("\nIl nome del programma corrisponde a argv[0], e vale \"%s\".\n\n",argv[0]);
  /* Nota: mettere un carattere '\\' prima del carattere '.' serve per poter
   * inserire il secondo all'interno della stringa da stampare */
  if (argc > 1)
  {
    printf("Il programma Ã.. stato chiamato con %d argomenti:\n\n", argc-1);
  }
}
```

```

for (ContaArgomenti = 1; ContaArgomenti < argc; ContaArgomenti++)
{
    printf("**** l'argomento argv[%d] e' la stringa \"%s\", composta da %d caratteri.\n", ContaArgomenti,
    argv[ContaArgomenti], strlen(argv[ContaArgomenti]));
    /* Nota: la compilazione di questa riga di programma genera un warning
    * ("format expects argument of type 'int', but argument 4 has
    * type 'size_t'") perché strlen restituisce un tipo di dato (size_t)
    * che è assimilabile a un int, ma non è un int. Questo warning può
    * essere tranquillamente ignorato. */
    if ('\0' == argv[ContaArgomenti][strlen(argv[ContaArgomenti])])
    {
        printf("La stampa di questo messaggio conferma che argv[%d] include un '\\0' di terminazione.\n", ContaArgomenti);
        /* Nota: mettere un carattere '\0' prima del carattere '\0' serve
        * per indicare che il secondo non e' parte di un carattere di
        * controllo (come '\n') ma va stampato così com'è */
    }
    printf("Ora tenterò di estrarre un float dall'argomento argv[%d] con la funzione sscanf... ", ContaArgomenti);
    Successo = sscanf(argv[ContaArgomenti], "%f", &ArgomentoFloat);
    /* Nota: sscanf (definita da stdio.h) si comporta come scanf, ma preleva
    * i dati da una stringa anziché da standard input. sscanf
    * restituisce un intero pari al numero di assegnamenti effettivamente
    * compiuti, ovvero sia di letture andate a buon fine (anche scanf
    * restituisce un intero determinato con la stessa regola; tuttavia
    * il valore restituito da scanf è usato raramente) */
    if (1 == Successo)
    {
        printf("L'estrazione ha avuto successo.\n");
        printf("Il dato estratto vale %f\n", ArgomentoFloat);
    }
    else
    {
        printf("L'estrazione non ha avuto successo.\n\n");
    }
}
else
{
    printf("Il programma e' stato chiamato senza argomenti.\n");
}

```

```

}
return 0;
}

```

```

/*****

```

Esempio di esecuzione:

```

$ ./prova abc d33 -1.234e5 y33.3e3tt 229

```

Gli elementi del comando appena inserito si trovano nell'array argv[].
Il nome del programma corrisponde a argv[0], e vale "./prova".

Il programma Ã stato chiamato con 5 argomenti:

```

*** l'argomento argv[1] e' la stringa "abc", composta da 3 caratteri.

```

(La stampa di questo messaggio conferma che argv[1] include un '\0' di terminazione.)

Ora tenterÃ di estrarre un float dall'argomento argv[1] con la funzione sscanf... L'estrazione non ha avuto successo.

```

*** l'argomento argv[2] e' la stringa "d33", composta da 3 caratteri.

```

(La stampa di questo messaggio conferma che argv[2] include un '\0' di terminazione.)

Ora tenterÃ di estrarre un float dall'argomento argv[2] con la funzione sscanf... L'estrazione non ha avuto successo.

```

*** l'argomento argv[3] e' la stringa "-1.234e5", composta da 8 caratteri.

```

(La stampa di questo messaggio conferma che argv[3] include un '\0' di terminazione.)

Ora tenterÃ di estrarre un float dall'argomento argv[3] con la funzione sscanf... L'estrazione ha avuto successo.

Il dato estratto vale -123400.000000

```

*** l'argomento argv[4] e' la stringa "y33.3e3tt", composta da 9 caratteri.

```

(La stampa di questo messaggio conferma che argv[4] include un '\0' di terminazione.)

Ora tenterÃ di estrarre un float dall'argomento argv[4] con la funzione sscanf... L'estrazione non ha avuto successo.

```

*** l'argomento argv[5] e' la stringa "229", composta da 3 caratteri.

```

(La stampa di questo messaggio conferma che argv[5] include un '\0' di terminazione.)

Ora tenterÃ di estrarre un float dall'argomento argv[5] con la funzione sscanf... L'estrazione ha avuto successo.

Il dato estratto vale 229.000000

```

*****/

```