

Complementi di Programmazione

Esercitazione 2

Tipi di dato primitivi

Esercizio 2.1

Definire una variabile per ogni tipo primitivo (char, short, int, long, float, double) e stampare la dimensione in byte di ciascuna di esse.

Esercizio 2.2

Date le seguenti variabili:

```
short s;  
int i;  
long l;  
float f;  
double d;
```

Scrivere un programma che prenda in input le suddette variabili e stampi il risultato delle seguenti espressioni:

1. $s+10 \cdot l$
2. $(s+i) \cdot l$
3. $(s+i) \cdot l + f$
4. $s/f + \sin(f)$
5. $l+1.5f$
6. $i < 10$
7. $d * 3.14159$

Nota: quando chiediamo di leggere un numero in input, scanf() automaticamente ignora eventuali caratteri bianchi iniziali ' ', '\n', '\t'. In particolare, viene ignorato il newline che abbiamo usato per terminare l'input precedente.

Esercizio 2.3

Scrivere un programma che stampi l'intero set dei caratteri ASCII, con la struttura:
"carattere" ; "codice carattere".

Esercizio 2.4

Scrivere un programma che calcoli il numero più grande possibile che una variabile di tipo `int` e una di tipo `long` possono immagazzinare.

N.B: ricorda che il tipi `long` e `int` comprendono il segno.

- A) Risolvere l'esercizio usando la libreria `<limits.h>`
- B) Calcolare il risultato aritmeticamente, senza usare la libreria `<limits.h>`

Puntatori

Esercizio 2.5

Completare il seguente programma in modo tale da assegnare alla variabile `c` il valore dell'espressione `a + b` utilizzando i puntatori `pa`, `pb` e `pc`. **Non è consentito** usare né assegnazioni dirette alla variabile `c` né operazioni aritmetiche dirette sulle variabili `a` e `b`.

```
int a = 10;
int b = -1;
int c;
int *pa, *pb, *pc;
// Assegnare i puntatori (da completare)
pa = ...
pb = ...
pc = ...
// Inserire codice qui (che contenga solo pa, pb, pc)
// ...
printf("%d == a + b\n", c);
```

Esercizio 2.6

Completare il seguente programma in modo tale da assegnare alla variabile `j` il valore della variabile `i` usando solo puntatori a `char` e senza usare l'istruzione di assegnamento tra interi (ad es., l'istruzione `j = i;` è proibita).

```
int i = 10;
int j = -1;
char *p, *q;
// Inserire codice qui (senza j = ...)
// ...
printf("%d == %d\n", i, j);
```

Esercizio 2.7

Definire due variabili intere `a` e `b`, calcolare la distanza `dist` in memoria tra queste variabili (tramite differenza di puntatori) e assegnare il valore 13 ad `a` utilizzando solo il puntatore `a` e `dist` (ovvero senza usare né `a` né il puntatore ad `a`).

Esercizio 2.8

Scrivere un programma che inizializzi in memoria un puntatore a intero `p`, ne determini il valore (valore dell'indirizzo) e scelga di conseguenza il tipo di una variabile, la cui allocazione in memoria sia di dimensione minima, che può contenere questo valore, scegliendo fra:

```
unsigned int
unsigned long int
unsigned long long int
```

Altri esercizi

Esercizio 2.9

Scopri il destino legato al tuo nome secondo la numerologia. I numeri del destino sono quelli compresi fra 1 e 9 più 11 e 22 (maggiori informazioni [qui](#)). Il numero del destino si ottiene sommando i codici ASCII delle lettere del nome e poi sommando le cifre di tale somma finché non viene un numero del destino. Scrivere un programma che legge uno alla volta i caratteri del nome e calcola il corrispondente numero del destino. Calcola il tuo numero.

Esercizio 2.10

Dati due interi i, j il cui valore è **preso da tastiera**, si calcoli il risultato della divisione $k = i/j$ di tipo `double`.

In seguito, si iteri sui primi decimali di k (massimo 10), ciascuno a distanza p rispetto alla virgola (la prima cifra decimale starà a distanza 0 dalla virgola), e **si stampi il carattere alfanumerico associato** in [ASCII](#) dopo aver aggiunto il corrispondente valore p .

Esempio:

Dati i seguenti valori di i, j, k :

```
int i=2
int j=3;
double k=i/j; // = 0.66666666
```

Il risultato sara':

'6'
'7'
'8'
'9'
'.'
'.'
'<'
'='
'>'
'?'

Esercizio 2.11

Si consideri il seguente calcolo:

```
float sum = 0;
for (int i = 0; i < 10; ++i)
{
    sum += 0.1f;
}
```

Si stampi il valore di `sum`, e si controlli se `(sum == 1.0f)`. In caso quest'ultimo test fallisca, si sostituisca l'uguaglianza con un test appropriato per confrontare numeri in virgola mobile.