

LIA - DEIS - Facoltà di Ingegneria - Università di Bologna
Laboratorio di informatica LA

Corso di laurea in Ingegneria Informatica - Anno Accademico 2007-2008

Input / Output, Espressioni Condizionali e Cicli

Esercizio 2.2

Si scriva un programma in linguaggio C, che legga giorno mese ed anno di una data e determini la data del giorno successivo (compreso il caso degli anni bisestili).

```
#include <stdio.h>

int main(void)
{
    int giorno, mese, anno;
    int giorni_del_mese;

    /* lettura della data */
    printf("Data (giorno, mese, anno) ? ");
    scanf("%d%d%d", &giorno, &mese, &anno);
    printf("\nGiorno successivo al %d/%d/%d e` ", giorno, mese, anno);

    /* calcola il numero dei giorni del mese */
    switch (mese) {
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            giorni_del_mese = 31;
            break;

        case 4: case 6: case 9: case 11:
            giorni_del_mese = 30;
            break;

        case 2:
            if (anno % 4 == 0 && anno != 1900)
                giorni_del_mese = 29;
            else
                giorni_del_mese = 28;
            break;
    }

    /* aggiorna la data del mese successivo */
    if (giorno == giorni_del_mese) {
        giorno = 1;
        if (mese == 12) {
            mese = 1;
            anno++;
        }
        else
            mese++;
    }
    else
```

Espressioni semplici

```
giorno++;

/* stampa la data aggiornata */
printf("%d/%d/%d\n", giorno, mese, anno);

return 0;
} /* main */
```

Esercizio 2.4

Si scriva un programma in linguaggio C, che proponga un menu con tre scelte possibili di cui una rappresenti la funzione di uscita dal programma.

```
#include <stdio.h>

main ()
{
    char carLetto;
    char inutile;

    printf ("*** MENU PRINCIPALE ***\n\n");
    printf ("scegli la tua opzione:\n");
    printf ("1: funzione 1\n");
    printf ("2: funzione 2\n");
    printf ("3: uscita\n\n");

    /* filtro l'input */

    carLetto = getch();

    switch (carLetto) {
        case '1': printf ("Sono nella funzione 1!\n");
                 printf ("\n\nPREMI UN TASTO PER CONTINUARE");
                 inutile = getch(); break;

        case '2': printf ("Hai scelto di premere 2\n");
                 printf ("\n\nPREMI UN TASTO PER CONTINUARE");
                 inutile = getch(); break;

        case '3': break;

        default: printf ("errore, qualcosa non va"); break;

    } // fine switch

    printf ("\n\nbuon proseguimento di lezione!\n");
}

}
```

Esercizio 2.5

Espressioni semplici

Si scriva un programma in linguaggio C, che legga da tastiera una sequenza di 0 e 1, terminata da 2 e calcoli la lunghezza della piu` lunga sottosequenza di 0, e la stampi.

ESEMPIO: 0 0 1 0 1 0 0 0 1 0 1 2 ---> 3.

```
#include <stdio.h>

int main(void)
{
    int bit;          /* l'elemento della sequenza appena letto */
    int cont = 0;     /* lunghezza attuale della sequenza di zeri;
                       inizialmente non abbiamo ancora letto nulla */
    int maxlung = 0;  /* valore temporaneo della massima lunghezza;
                       inizialmente e` 0 */

    printf("Inserisci una sequenza di 0, 1 terminata da 2\n");
    do {
        scanf("%d", &bit);
        if (bit == 0) {
            cont++;          /* e` stato letto un altro 0 */
            /* aggiorna la lunghezza della sequenza corrente */
            if (cont > maxlung) /* se necessario, aggiorna il massimo temporaneo*/
                maxlung = cont;
            /* alternativa piu` costosa: maxlung := maxlung +1 */
        }
        else                /* e` stato letto un 1 (o il 2 finale) */
            cont = 0;       /* azzera la lunghezza della sequenza corrente */
    } while (bit != 2);    /* quando si legge 2 il ciclo termina */

    printf("La piu` lunga sottosequenza di soli zeri e` lunga %d\n", maxlung);

    return 0;
}
```

Esercizio 3.1 (for annidati)

Si scriva un programma in linguaggio C che legga un intero compreso fra 1 e 9 e

Ad esempio, se legge 9, stampa quanto segue:

```
1
121
12321
1234321
123454321
12345654321
1234567654321
123456787654321
12345678987654321
```

Soluzione

```
#include <stdio.h>

int main(void)
```

Espressioni semplici

```
{
    int altezza, riga, i;

    printf("Altezza piramide (1..9) ? ");
    scanf("%d", &altezza);

    for (riga = 1; riga <= altezza; riga++) {
        /* tiene conto delle righe della piramide: 1 iterazione <-> 1 riga */
        for (i = 1; i <= altezza - riga; i++)
            /* stampa gli spazi bianchi iniziali necessari in ogni riga */
            putchar(' ');

        for (i = 1; i <= riga; i++)
            /* stampa la sequenza crescente di numeri di ogni riga */
            printf("%d", i);

        for (i = riga - 1; i >= 1; i--)
            /* stampa la sequenza decrescente di numeri di ogni riga */
            printf("%d", i);

        putchar('\n');
    }
    /* va a capo: la riga e' finita */

    return 0;
}
```

Esercizio 3.2 (ciclo for annidato caso in cui il numero di interazioni del ciclo più interno non dipende dall'interazione del ciclo più esterno)

Si scriva un programma in linguaggio C che stampi una tavola pitagorica con rig

Soluzione

```
#include <stdio.h>

#define Nmax 10

int main(void)
{
    int riga, colonna;
    int i;

    /* stampa della riga di intestazione */
    printf("%5c", ' ');
    for (colonna = 1; colonna <= Nmax; colonna++)
        printf("%4d", colonna);
    putchar('\n');
    for (i = 1; i <= Nmax * 4 + 5; i++)
        putchar('-');
    putchar('\n');

    for (riga = 1; riga <= Nmax; riga++) {
        printf("%2d | ", riga);
        for (colonna = 1; colonna <= Nmax; colonna++)
            printf("%4d", riga * colonna);
        putchar('\n');
    }
}
```

Espressioni semplici

```
    return 0;  
}
```