

LIA - DEIS - Facoltà di Ingegneria - Università di Bologna
Elementi di informatica L

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

Funzioni

Esercizio 3.3

Si scriva un programma in linguaggio C, che calcoli il minimo comune multiplo e il massimo comun divisore di due interi forniti da input. (si usino due funzioni per la determinazione del minimo comune multiplo e del massimo comun divisore)

Soluzione

```
#include <stdio.h>

int massimoComunDivisore(int n1, int n2);

int minimoComuneMultiplo(int n1, int n2);

int main(void)
{
    int num1, num2;
    int mcm, mcd;

    printf("Inserire due numeri interi: ");
    scanf("%d%d", &num1, &num2);

    /* chiamata della funzione minimoComuneMultiplo */
    mcm = minimoComuneMultiplo(num1, num2);

    /* chiamata della funzione massimoComunDivisore */
    mcd = massimoComunDivisore(num1, num2);

    printf("\nIl minimo comune multiplo tra %d e %d e` %d\n", num1, num2, mcm);
    printf("\nIl massimo comun divisore tra %d e %d e` %d\n", num1, num2, mcd);

    scanf("%*[^\\n]"); getchar();
    printf("\nPremi [INVIO] per terminare\n");
    getchar();

    return 0;
}

int minimoComuneMultiplo(int n1, int n2)
{
    return n1 * n2 / massimoComunDivisore(n1, n2);
}
```

LIA

```
int massimoComunDivisore (int n1, int n2)
/* calcola il massimo comun divisore di due interi * usando il metodo di
   Euclide */
{
    int resto, a, b;

    a = n1;
    b = n2;
    while (b > 0) {
        resto = a % b;
        a = b;
        b = resto;
    }

    return a;
}
```

Esercizio 3.4 (funzioni e uso del "switch - case")

Si scriva un programma in linguaggio C che legge da tastiera una sequenza di caratteri. Si assuma che la sequenza di caratteri rappresenti un numero romano corretto.

In particolare, si ammetta che vi sia una sottosequenza di due "cifre" in ordine.

Soluzione

```
#include <stdio.h>
```

```
int Romano2Intero(char ch)
{
    int ris;

    switch (ch) {
        case 'I': ris = 1; break;
        case 'V': ris = 5; break;
        case 'X': ris = 10; break;
        case 'L': ris = 50; break;
        case 'C': ris = 100; break;
        case 'D': ris = 500; break;
        case 'M': ris = 1000; break;
        default: ris = -1;
    }

    return ris;
}
```

```
int main(void)
{
    char corr, succ;
    int val_corr, val_succ;
    int somma = 0;

    printf("Inserisci un numero romano (senza spazi) terminato da '\\n\\n");
    printf("Il numero romano deve essere corretto!\\n");
    printf("(cioe' una sequenza di cifre M,D,C,L,X,V,I in ordine corretto)\\n");
    scanf("%c", &corr);
}
```

```

if (corr != '\n') {
    val_corr = Romano2Intero(corr);
    scanf("%c", &succ);
    while (succ != '\n') {
        val_succ = Romano2Intero(succ);
        if (val_corr >= val_succ)
            somma = somma + val_corr;
        else somma = somma - val_corr;
        corr = succ;
        val_corr = val_succ;
        scanf("%c", &succ);
    }
    somma = somma + val_corr;
}
printf("Numero corrispondente: %d\n", somma);

return 0;
}

```

Esercizio 3.5 (funzioni con controllo dell'input)

Codificare in C una funzione che, dato il valore (da 1 a 12) assunto da una variabile *mese*, restituisca il numero di giorni corrispondenti. Si assume che l'anno sia non bisestile.

Soluzione

```

#include <stdio.h>

int GiorniMese(int mese)
{
    if (mese == 2) return 28; /* di ventotto ce n'e' uno... */
    else if ((mese==11) || (mese==4) || (mese==6) || (mese==9)) return 30;
    /* 30 ne ha novembre... */
    else return 31; /* tutti gli altri ne han 31 */
}

main()
{
    int mese;

    do{
        printf("Inserire il numero corrispondente al mese: ");
        scanf("%d",&mese); /* aggiungere la diagnostica degli errori */
    }while (mese<1 || mese>12);
    /* fintanto che il numero inserito non e' compreso tra 1 e 12 si ripete
    la lettura */

    printf("Il mese %d ha %d giorni",mese,GiorniMese(mese));
}
/* notare l'utilizzo diretto del risultato della funzione GiorniMese() */

```

LIA

- la larghezza
- lo spostamento a destra

```
#include <stdio.h>

void StampaRiga(int spazi, int lung) /* PARAMETRI FORMALI */
{
    /* VARIABILI LOCALI */
    int i; /* indice del ciclo che scandisce le posizioni */

    for (i = 1; i <= spazi; i++)
        putchar(' ');
    for (i = 1; i <= lung; i++)
        putchar('*');
    putchar('\n');
} /* StampaRiga */

int main(void)
{
    int altezza; /* altezza del rettangolo */
    int larghezza; /* larghezza del rettangolo */
    int spostamento; /* spostamento a destra */
    int i; /* indice del ciclo che scandisce le righe */

    printf("Altezza rettangolo? ");
    scanf("%d", &altezza);
    printf("Larghezza rettangolo? ");
    scanf("%d", &larghezza);
    printf("Spostamento a destra? ");
    scanf("%d", &spostamento);
    for (i = 1; i <= altezza; i++) {
        StampaRiga(spostamento, larghezza); /* spostamento e larghezza sono i
                                           PARAMETRI ATTUALI corrispondenti ai
                                           parametri formali spazi e lung */
    }

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