Consorzio Nettuno - Ingegneria Inf., El., Tel.
Fond. Di Informatica 1 – Tecniche della Programmazione 1° modulo
22-5-2010

Esercizio 1

Sia data una matrice quadrata di interi di dimensioni N*N ed una seconda matrice N*N.
Si scriva la funzione C che copia la prima matrice nella seconda matrice ruotandola di un quarto di giro rispetto al centro. (quindi ad esempio porta l'elemento a coordinate (0,0) nella posizione (N-1,0), quello a coordinate (N-1,0) nella posizione (N-1,N-1) e così via)
Esempio

1	2	3
4	5	6
7	8	9

diventa

7	4	1
8	5	2
9	6	3

Soluzione

Dovendo solo copiare gli elementi in una nuova matrice non c'è bisogno di elementi temporanei per fare scambi, ma l'esercizio si risolve facendo attenzione alle coordinate di destinazione degli elementi. In pratica ciascuna riga della prima matrice dev'essere copiata in una colonna della seconda matrice, la prima riga nell'ultima colonna, la seconda riga nella penultima e così via, ovvero l'elemento a coordinate (R,C) dev'essere copiato nell'elemento a coordinate (N-1-C,R).

```
#define N 3
void ruota(int M1[N][N], int M2[N][N]) {
     int riga, colonna;
     for (riga = 0 ; riga < N ; riga ++)
          for (colonna = 0 ; colonna < N ; colonna ++)
               M2[N - 1 - colonna][riga] = M1[riga][colonna];
}
```

Esercizio 2

Si scriva la funzione C che riceve come argomenti una stringa contenente un testo ed un numero intero, e che deve cercare se nel testo è presente quel numero in formato testuale.
Come risultato la funzione deve tornarne la posizione della prima cifra, oppure il valore -1 se il numero non viene trovato.
Esempio
se gli argomenti sono: "la vecchia con la borsa salta 42 fossi senza rincorsa"
ed il numero 42, il risultato sarà 30.

Soluzione

Per cercare nella stringa testo il numero bisogna prima trasformare il numero nella corrispondente forma testuale, poi basta cercare la stringa ottenuta nel testo. La conversione di un numero in forma testuale può essere svolta con la funzione sprinf oppure calcolando una cifra per volta e costruendo la stringa corrispondente.

Prima opzione, trasformare il numero in stringa con sprintf.

```
#include <string.h>
```

```
int cercaNumero(int numero, char * testo) {
     int pos, car;
     char daCercare[100] = { 0 };           // faccio un buffer
     sprintf(daCercare, "%d", numero);      // converto il numero
     // scandisco le posizioni possibili nel testo
     for (pos=0 ; testo[pos] != '\0' ; pos++) {
          // per ogni posizione scandisco i caratteri da cercare
          for (car=0 ; daCercare[car] != '\0' ; car++)
               // se il carattere è diverso passo alla prossima pos
               if (daCercare[car] != testo[pos+car])
                    break;
          // se arrivo qui tutti i caratteri erano uguali, trovato
          return pos;
     }
     // altrimenti non è stato trovato
     return -1;
}
```

Seconda opzione, convertendo il numero un carattere per volta (mostro solo la conversione)

```
void convertiNumero(int numero, char * buffer) {
     // converto una cifra per volta e la copio nel buffer
     // e rovescio la stringa per avere le cifre nel giusto ordine
     int i = 0, j, cifra, resto = numero;
     char tmp;
     while (resto != 0) {
          cifra = resto % 10;      // calcolo la cifra delle unità
          resto /= 10;             // e divido per 10
          buffer[i] = '0' + cifra; // e ottengo il carattere
          i++;
     }
     // i ora contiene il numero di cifre, rovescio la stringa
     for ( j=i-1, i=0 ; i < j ; i++,j-- ) {
          tmp = buffer[i];
          buffer[i] = buffer[j];
          buffer[j] = tmp;
     }
}
```

Esercizio 3

Si scriva la funzione C che conta il numero di 1 che sono presenti nella codifica binaria di un carattere (ad esempio A=01000001 => 2).
Usando la precedente funzione scrivete il programma che chiede una stringa e che stampa il numero di caratteri della stringa che hanno un numero pari di 1 (in grassetto nell'esempio).

Esempio: "**AB**C**D**EF**GH**I" risultato => 5
Infatti la codifica binaria di A è 01000001, B=01000010, C=01000011 ...

Soluzione

```
int contaUni(char cifra) {
     int quanti = 0, i;
     for (i=0 ; i<8 ; i++) {       // calcolo le cifre binarie
          quanti += cifra % 2;     // e le sommo
```

```
          cifra /= 2;
     }
     return quanti;
}
int quantiPari( char * testo ) {
     int i, quanti=0;
     for (i=0 ; testo[i] != '\0')       // per ogni carattere
          if (contaUni(testo[i]) % 2 == 0)   // se uni sono pari
               quanti++;                // lo conto
     return quanti;
}
```

Esercizio 4

Si scriva il programma che indovina il numero pensato dall'utente (tra 0 e 100000), proponendo dei tentativi secondo il metodo della ricerca binaria.
L'utente ad ogni tentativo può rispondere
0 = trovato
1 = tentativo troppo alto
-1 = tentativo troppo basso

Soluzione

Si deve realizzare la ricerca binaria usando la risposta dell'utente come indicazione se il numero tentato è maggiore, uguale o minore del numero da trovare.

All'inizio l'intervallo di valori possibili è da 0 a 100000.

La ricerca binaria si fa proponendo via via il numero di mezzo dell'intervallo di valori ancora possibili, ed aggiornando l'intervallo a seconda della risposta.

Per rappresentare un intervallo è sufficiente ricordarne l'inizio e la fine.

```
#include <stdio.h>
int main() {
     int inizio = 0, fine = 100000, mezzo, risposta;
     while (inizio < fine) {            // finchè ci sono numeri
          mezzo = inizio + fine / 2;         // calcolo il mezzo
          printf("Provo con %d\n", mezzo);   // lo propongo
          scanf("%d", &risposta);            // e chiedo
          if (risposta == 0) {
               printf("indovinato!!!";
               return 0;
          }
          if (risposta < 0) {           // il numero è maggiore
               printf("basso!!!";
               inizio = mezzo + 1;
          }
          if (risposta > 0) {           // il numero è minore
               printf("alto!!!";
               fine = mezzo - 1;
          }
     }
     printf("hai barato!!!");
}
```