

```
/*  
 * Corso di Fondamenti di Informatica  
 * Esercizio:  
 * Esempio semplice di elaborazione array  
 */
```

```
/*  
*****  
Scrivere un programma C che compie le seguenti operazioni:  
*****
```

1. Stampa a schermo un messaggio che chiede all'utente di inserire un numero intero.
  2. Se il numero inserito è negativo o nullo, lo memorizza in uno degli elementi di un array di interi avente DIMENSIONE\_ARRAY (ad esempio 10) elementi, dove DIMENSIONE\_ARRAY è un parametro definito con una direttiva #define.
  3. Se il numero inserito è positivo o se tutti gli elementi dell'array sono stati usati termina la richiesta di numeri, altrimenti torna al punto 1.
- NOTA: si supponga che l'utente inserisca sempre almeno un numero negativo o nullo.

4. Stampa a schermo la somma di tutti i numeri interi  $\leq 0$  forniti dall'utente (attenzione: NON di tutti i numeri contenuti nell'array!).
5. Stampa a schermo un messaggio che indica se tra i numeri  $\leq 0$  inseriti sono in maggioranza quelli pari, quelli dispari o sono in uguale numero (suggerimento: l'operatore % dà il resto della divisione tra interi).
6. Chiede all'utente di inserire un numero reale e lo memorizza.
7. Per ciascuno dei numeri interi inseriti in precedenza dall'utente, stampa a schermo il risultato della divisione del numero reale (dividendo) per il numero intero (divisore). Se il divisore è zero, al posto del risultato stampa un messaggio che segnala il fatto, SENZA eseguire la divisione. La stampa deve avere la forma "23.456 / -76 = -0.30863157", in modo da mostrare esplicitamente i valori di dividendo e divisore.
8. Trova il minimo tra i numeri interi  $\leq 0$  inseriti dall'utente e lo stampa a schermo. Se poi tra i numeri forniti dall'utente ne esiste più di uno pari al minimo, stampa un ulteriore messaggio per segnalare il fatto.

```
*****  
#include <stdio.h>  
  
#define DIMENSIONE_ARRAY 10  
/* dimensione dell'array destinato a contenere gli interi forniti  
 * dall'utente; deve essere almeno pari ad 1 */  
  
int main()  
{  
    int NumeriInteri[DIMENSIONE_ARRAY];  
    /* contiene gli interi forniti dall'utente */  
    int Input;  
    /* numero intero fornito dall'utente */  
    int NumDataInseriti;  
    /* numero degli interi negativi o nulli inseriti dall'utente */  
    int Indice;  
    /* tiene conto della posizione raggiunta durante la scansione dell'array */  
    int Somma;  
    /* somma parziale degli elementi dell'array */  
    int Pari, Dispari;  
    /* numero di dati <=0 pari [dispari] inseriti dall'utente */  
    float Dividendo;  
    /* contiene il numero reale da dividere per ciascuno degli interi */  
    int Indiceminimo;  
    /* contiene l'indice nell'array del valore minimo tra quelli trovati  
     * esaminando l'array */  
    int MinimoDuplicato;  
    /* vale 1 se esistono più numeri pari al valore minimo, 0 altrimenti */  
  
    /** Fase di input degli interi **/  
    printf("\nInserisci un intero <=0 [1/%d]: ", DIMENSIONE_ARRAY);  
    scanf("%d", &Input);  
  
    Indice = 0;  
    while ((Input <= 0) && (Indice < DIMENSIONE_ARRAY))  
    {  
        NumeriInteri[Indice] = Input;  
        ++Indice;  
    }  
}
```

```
printf("\nInserisci un intero <=0 (se >0 l'input termina) [%d/%d]: ",
      Indice+1, DIMENSIONE_ARRAY);
scanf("%d", &Input);
}

NumDatiiInseriti = Indice;
printf("\nFase di input terminata.");

/* ***** */
/* codice usato in fase di debugging, ma lasciato come esempio */
printf("\n\nContenuto dell'array:");
Indice = 0;
while(Indice < NumDatiiInseriti)
{
    printf("\t%d", NumeriInteri[Indice]);
    ++Indice;
}
/* ***** */

/** Calcolo della somma e del suo valore assoluto ***/
Indice = 0;
Somma = 0;
while (Indice < NumDatiiInseriti)
{
    Somma = Somma + NumeriInteri[Indice];
    ++Indice;
}

printf("\n\nSomma dei numeri interi: %d", Somma);

/** Determinazione dei numeri pari e dispari ***/
Pari = 0;
Dispari = 0;
Indice = 0;
while(Indice < NumDatiiInseriti)
{
    if (0 == NumeriInteri[Indice]%2)
```

```
        {
            ++Pari;
        }
    else
    {
        ++Dispari;
    }

    ++Indice;
}

if (Pari == Dispari)
{
    printf("\nNumeri pari e dispari in uguale numero.");
}
else
{
    printf("\nSono in maggioranza i numeri ");
    if (Pari>Dispari)
    {
        printf("pari.");
    }
    else
    {
        printf("dispari.");
    }
}

/** Calcolo delle divisioni */
printf("\n\nInserisci un numero reale: ");
scanf("%f", &Dividendo);

Indice = 0;

while(Indice < NumDatInseriti)
{
    printf("\n%f / %d = ", Dividendo, NumeriInteri[Indice]);
    if (0 != NumeriInteri[Indice])
    {
        printf("%f", Dividendo/NumeriInteri[Indice]);
    }
}
```

```
    }  
    else  
    {  
        printf("divisione per zero!");  
    }  
    ++Indice;  
}  
  
/** Ricerca del minimo */  
/* cominciamo supponendo che il numero minimo sia quello nella prima cella  
 * dell'array; poi confrontiamo tale numero con ciascuno degli altri elementi  
 * dell'array: ogni volta che troviamo un numero più basso del precedente lo  
 * prendiamo come nuovo minimo */  
IndiceMinimo = 0;  
Indice = 1;  
while (Indice < NumDatiInseriti)  
{  
    if (NumeriInteri[Indice] < NumeriInteri[IndiceMinimo])  
    {  
        IndiceMinimo = Indice;  
    }  
    ++Indice;  
}  
  
printf("\n\nIl valore minimo e': %d.", NumeriInteri[IndiceMinimo]);  
  
/** Ricerca dell'eventuale duplicato del valore minimo */  
Indice = 0;  
MinimoDuplicato = 0;  
while (Indice < NumDatiInseriti)  
{  
    if ( (NumeriInteri[Indice] == NumeriInteri[IndiceMinimo]) &&  
        (IndiceMinimo != Indice) )  
    {  
        MinimoDuplicato = 1;  
    }  
    ++Indice;  
}
```

```
printf("\nIl valore minimo ");  
if (0 == Minimoduplicato)  
{  
    printf("non ");  
}  
printf("compare piu' volte tra i numeri inseriti dall'utente.\n\n");  
return(0);  
}
```