


INFORMATICA ED ELEMENTI DI INFORMATICA MEDICA

MODULO DI INFORMATICA
Prof. Stefano Gaburri e Prof. Gianluca Palermo

Prova del 28 Giugno 2010

	Nome:	Cognome:
	Matricola:	Firma:

- **Non separate questi fogli.** Scrivete la soluzione **solo sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità. **Cancellate le parti di brutta** (o ripudiate) con un tratto di **penna**.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **È possibile scrivere a matita** (e non occorre ricalcare al momento della consegna!).
- È **vietato** utilizzare **calcolatrici, telefoni, pc, libri e appunti**. Chi tenti di farlo vedrà **annullata** la sua prova.
- Qualsiasi **tentativo** di comunicare con altri studenti comporta **l'espulsione** dall'aula.
- È possibile **ritirarsi senza penalità**.
- Non è possibile lasciare l'aula conservando il tema della prova in corso.

- Il presente plico contiene **3 esercizi** e **7 pagine** compresa questa, deve essere debitamente compilato con cognome e nome, numero di matricola.
- Il tempo a disposizione è di **1ora e 30minuti**.

E' OBBLIGATORIO SCRIVERE IN MANIERA LEGGIBILE, LE PARTI DEL COMPITO NON COMPRESIBILI AL DOCENTE VERRANNO CONSIDERATE ERRATE!

ESERCIZIO 1 – Comprensione del codice

Si consideri il seguente programma C

```
#include <stdio.h>
#define MAX 10

int main()
{
    typedef int numeri[MAX];
    numeri A;
    int p,j,i=0,s=0;
    int finito = 0;

    do {
        scanf("%d", &A[i]);
        if (A[i]>0) i++;
        else finito = 1;
    } while(i<MAX && finito==0);

    do {
        scanf("%d",&p);
    } while(p<0 || p>=i);

    for(j=i-1;j>=p;j--)
        s+=A[j];

    for(j=0;j<p;j++)
        s-=A[j];

    if(s>0)
        printf("-sol- %d\n",s);
    else
        printf("-sol- %d\n",-s);
}
```

1) Si aggiungano brevi commenti (massimo 4 righe per commento!) per spiegare CHIARAMENTE cosa fa ogni ciclo e il test finale.

```
#include <stdio.h>
#define MAX 10

int main()
{
    typedef int numeri[MAX];
    numeri A;
    int p,j,i=0,s=0;
    int finito = 0;

    do {
        scanf("%d", &A[i]);
        if (A[i]>0) i++;
        else finito = 1;
    } while (i<MAX && finito==0);
    //
    //
    //
    //
    //
    do {
        scanf("%d",&p);
    } while(p<0 || p>=i);
    //
    //
    //
    //
    //
    for(j=i-1;j>=p;j--)
        s+=A[j];
    //
    //
    //
    //
    //
    for(j=0;j<p;j++)
        s-=A[j];
    //
    //
    //
    //
    //
    if(s>0)
        printf("-sol- %d\n",s);
    else
        printf("-sol- %d\n",-s);
    //
    //
    //
    //
    //
}
```

2) Si indichi cosa viene stampato a video se l'utente inserisce questa sequenza, giustificando il risultato finale:

2 5 6 7 5 4 2 1 0 3

Soluzione:

-sol- 6

Il programma acquisisce in A un insieme di numeri >0 , al primo numero minore o uguale a 0 interrompe l'acquisizione

Il numero successivo è usato per indicare quanti dei primi numeri acquisiti dell'insieme A devono essere sottratti alla somma dei restanti numeri dello stesso insieme.

3) Che cosa succederebbe se l'utente invece di 3 inserisse -1 come ultimo valore della sequenza?

Soluzione:

Non si uscirebbe dal /*Ciclo 2*/ e si aspetterebbe un altro numero dall'utente.

ESERCIZIO 2 - Le funzioni

Negli ospedali Riuniti di Capitello sull'Aglio è necessario progettare un sistema che gestisca informazioni riguardanti i pazienti ricoverati. L'ospedale ha solo dodici posti letto e non è possibile avere in alcun modo un numero di ricoverati superiori alla ricettività della struttura.

1) Dichiarare due strutture dati:

- una di tipo `tPaziente` che serve per la gestione di una semplice cartella del paziente che includa nome, cognome, codice fiscale, diagnosi, codice medico di riferimento.

```
typedef struct { stringa nome;  
                stringa cognome;  
                stringa codiceFiscale;  
                stringa diagnosi;  
                int codiceMedico;  
            } tPaziente;
```

- una di tipo `tPostiLetto` che serve per tenere traccia dello stato dei posti letto in ospedale e dell'eventuale paziente associato.

Versione1

```
typedef struct { tPaziente paziente[12];  
                int libero[12];  
            } tPostiLetto;
```

oppure

Versione2

```
typedef struct { tPaziente paziente;  
                int libero;  
            } tLetto;
```

```
typedef tLetto tPostiLetto[12]
```

2) Scrivete i soli prototipi delle seguenti due funzioni C:

- a) la prima di nome "modificaDati" modifica i dati di un paziente all'interno della sua cartella (di tipo `tPaziente`) passata come parametro. I dati riguardanti la modifica sono acquisiti all'interno della funzione stessa.

```
void modificaDati(tPaziente *nome);
```

- b) la seconda di nome "assegnaLetto" prende in ingresso la cartella di un paziente e la inserisce nel primo posto letto a disposizione nell'ospedale. La struttura che gestisce i posti letto (`tPostiLetto`) è anch'essa passata come parametro. La funzione restituirà 0 (zero) in caso di nessun posto letto disponibile.

```
int assegnaLetto(tPaziente *cartella, tPostiLetto * ospedale);
```

3) Implementare la funzione “assegnaLetto” definita al punto 2b

Una delle possibili soluzioni considerando Versione1 della struttura tPostiLetto:

```
int assegnaLetto(tPaziente *cartella, tPostiLetto * ospedale){  
  
    int i;  
    for (i=0; i<12; i++){  
        if (ospedale.libero[i]==0){  
            ospedale.paziente[i]=(*cartella);  
            ospedale.libero[i]=1;  
            return 1;  
        }  
    }  
    return 0;  
}
```

ESERCIZIO 3 - Ricorsione

Un noto esperto di bellezze canine è stato nominato giudice unico del concorso PoliDOG che eleggerà il cane più bello dei dipendenti del Politecnico di Milano. Come si confà all'occasione, i cani sono rappresentati dalla seguente struttura dati:

```
typedef struct { stringa nome; // nome del cane
                int voto;      // quanto è bello il cane?
            } cane
```

I cani sono organizzati in un gruppo di tipo `t_gruppo` di cui non conosciamo la struttura interna, però sappiamo che sono definite le seguenti funzioni:

```
int contaCani (t_gruppo g);
// restituisce il numero di cani nel gruppo

stringa giudicaCani (t_gruppo g);
// funziona SOLO se sono rimasti in lizza uno o due cani
// restituisce il nome del più bello dei 2 (o dell'unico presente)

void selezionaBelli (t_gruppo originale, t_gruppo * belli);
// seleziona dal gruppo originale solo i cani più belli.
// del gruppo originale sarà scartato almeno un cane.
```

- Scrivere la funzione ricorsiva che prende un gruppo di cani e restituisce il nome del vincitore.

Soluzione:

```
stringa eleggiPoliDOG (t_gruppo candidati) {
t_gruppo selezionati;

if (contaCani(candidati) <= 2) return giudicaCani (candidati);
else { selezionaBelli (candidati, &selezionati);
      return eleggiPoliDOG (selezionati); }
}
```