

Politecnico di Milano - Anno Accademico 2001-2002 - Informatica B
Recupero – 25 Febbraio 2002 – Elaborato A

NOME COGNOME	
Matricola	

REC2 /Punti	I / 2 punti	II / 2 punti	III / 2 punti	IV /3 punti	V /3 punti	REC1
Tot= / 12						

REC2 I – Calcolatore – Completare un semplice programma in C e in Assembler che somma 3 numeri già contenuti in una tabella NUMERI e scrive il risultato. Usare le frasi già predisposte (per semplificare; si consiglia di **NON stravolgere** l'impostazione) (2 punti)

	<i>Assembler</i>	<i>Linguaggio C</i>
EXIT	EQU 2000	#define LMAX 3
LMAX	EQU 3	Void main()
PORTA_Video	EQU 2	
CNT	DW ?	int CNT;
NUMERI	DW 2	int NUMERI[LMAX]={2,5,8};
	DW 5	
	DW 8	
SOM	DW ?	Int SOM ;
START	LDA #0	{CNT = 0 ;
	STA CNT	
	STA SOM	SOM = 0 ;
CALCOLA	LDA CNT	while (CNT != LMAX)
	SUB #LMAX	{
	JZ SCRIVI	
	LDA SOM	SOM = SOM + NUMERI[CNT];
	LDI CNT	
	ADD NUMERI(I)	
	STA SOM	
	LDA CNT	CNT=CNT+1;
	ADD #1	
	STA CNT	
	JMP CALCOLA	} * end while
SCRIVI	LDA SOM	Scrivi SOM ;
	OUT PORTA_Video	
	JMP EXIT	} * end main

Politecnico di Milano - Anno Accademico 2001-2002 - Informatica B
Recupero – 25 Febbraio 2002 – Elaborato A

NOME COGNOME

Matricola

REC2 - II – cicli e funzioni (2 punti)

Data la seguente definizione di struttura

```
typedef struct {int a; char b[7];}ST;
```

la funzione che segue serve a caricare una struttura di tipo ST:

```
ST caricaF(int c, char *b)
```

```
{  
    ST aux;  
    aux.a=c;  
    strcpy(aux.b,b);  
    return(aux);}
```

Si chiede di

1. trasformare la **funzione** ST caricaF(int c, char *b) nella **procedura** equivalente void carica P(int c, char *b, ..?..completare..) e di codificarla.

```
void caricaP(int c, char *b, ST *el)  
{ el->a=c;  
  strcpy(el->b,b);  
}
```

2. scrivere il frammento di codice che, richiamando la funzione caricaF(), permette di caricare i dieci elementi della variabile ST vt[10] leggendo i necessari dati da tastiera

```
ST vt[10];  
int i, a;  
char s[7];  
for(i=0;i<10;i++)  
{scanf("%d",&a);  
  gets(s);  
  vt[i] = caricaF(a,s);}
```

REC2 - III – Files (2 punti)

Il **file binario** inp.dat contiene strutture (in quantità non nota a priori) del seguente tipo:

```
typedef struct {char s[9]; int a; double b;}EL;
```

Il programma da completare, sotto riportato, deve leggere da inp.dat tali strutture e generare in output il **file di testo** exit.txt le cui righe sono formate dal contenuto del campo s della struttura seguito da ":" quindi dalla stringa che rappresenta il campo a, anch'essa seguita da ":" ed, infine, dalla stringa che rappresenta il campo b. Per esempio partendo da una struttura con i campi contenenti i seguenti valori:

s	a	b
ciao	46	101.25

si pone nel file di testo:

Politecnico di Milano - Anno Accademico 2001-2002 - Informatica B
Recupero – 25 Febbraio 2002 – Elaborato A

NOME COGNOME	
Matricola	

ciao:46:101.25

Parimenti da

s	a	b
prova	15	99.41

si pone nel file di testo:

prova:15:99.41

Potrà essere utile l'uso della funzione `fprintf()`. Si supponga già caricato il file `inp.dat`.

```
main()
{FILE *fin, *fout;
  EL elem;

  /* apertura files - completare! */
  fin=fopen("inp.dat","rb");
  fout=fopen("exit.txt","w");

  if((fin!=NULL) && (fout!=NULL))
  { /* ciclo lettura-scrittura fino a esaurimento del file fin - completare! */
    fread(&elem, sizeof(elem), 1, fin);
    while(!feof(fin))
      { fprintf(fout,"%s:%d:%lf\n",elem.s,elem.a,elem.b);
        fread(&elem, sizeof(elem), 1, fin);}
    fclose(fin); fclose(fout);
  }
  else
  { /* segnalazione errore - completare! */
    printf("errore apertura file!\n");
  }
  return(0);
}
```

REC2 - IV – liste (3 punti)

Data una lista, che potete supporre contenga almeno 4 elementi, basata sulla seguente definizione:

```
struct nodo_s { int valore; struct nodo_s * next; }
struct nodo_s * head;
```

scrivere un frammento di codice che assegna 75 al valore contenuto nel terzo nodo della lista solo se il valore contenuto nel primo nodo è maggiore del valore contenuto nel secondo nodo

```
if(head->valore > head->next->valore)
    head->next->next->valore=75;
```

scrivere le istruzioni necessarie per eliminare il primo nodo della lista

NOME COGNOME	
Matricola	

```
struct nodo_s *tmp=head;  
head=head->next;  
free(tmp);
```

REC2 - V – Programma (3 punti)

Un modo per approssimare i valori della funzione logaritmo naturale è dato dalla seguente serie:

$$\ln x = 2 \left[a + \frac{1}{3}a^3 + \frac{1}{5}a^5 + \frac{1}{7}a^7 + \dots \right] \quad \text{dove } a = \frac{x-1}{x+1}$$

il valore del logaritmo così calcolato è tanto più preciso quanti più termini della serie si sommano; ci si può quindi limitare a prendere in considerazione tanti termini quanti bastano per ottenere la precisione desiderata.

Per esempio, ci si può fermare quando si riscontra che la differenza fra il valore ottenuto sommando K termini e il valore ottenuto sommando K+1 termini non supera, in valore assoluto, un valore desiderato (cioè la precisione richiesta).

Scrivere il programma che, facendo riferimento alla formula sopra indicata ed alle istruzioni fornite, calcola il valore del logaritmo per un x inserito da tastiera, approssimando il risultato alla seconda cifra decimale (ovvero in modo che la differenza fra due valori consecutivi della serie troncata sia minore in valore assoluto di 0,001).

Può essere utile usare la funzione `double fabs(double val)` che dato un valore ne restituisce il valore assoluto e, ma non è indispensabile, `double pow(double base, double esponente)` che restituisce il valore $\text{base}^{\text{esponente}}$. Saranno opportunamente valutate le soluzioni più efficienti.

```
double logx(double x)  
{ int passo=1;  
  double prec=0.0, att=0.0, rapp, pot=1.0;  
  rapp=(x-1)/(x+1);  
  att=rapp;  
  while(fabs(2*att-2*prec)>0.001)  
  { prec=att;  
    pot=pot*rapp*rapp;  
    att = att + pot/passo;  
    passo=passo+2;  
  }  
  return(prec);  
}
```

Politecnico di Milano - Anno Accademico 2001-2002 - Informatica B
Recupero – 25 Febbraio 2002 – **Elaborato A**

NOME COGNOME

Matricola

REC1 – Recupero Primo Compitino

Programmare una pagina HTML che appaia come in figura. (usare le stesse frasi e raggiungere lo stesso effetto)

1. il titolo della pagina sia **Primo Esercizio di Recupero**
2. la pagina ha una intestazione di massima grandezza
3. una tabella centrata ha una riga di intestazione e due righe di contenuti
4. in ogni riga le tre colonne contengono
 - a. un link ad una pagina che descrive il quadro sul sito www.ggo.com (ogni quadro ha una pagina diversa del tipo "titolo.html")
 - b. l'immagine del quadro "titolo.gif"
 - c. il prezzo in Euro (inserire anche il simbolo dell'Euro &euro)
5. sotto la tabella una riga lampeggiante ed in *italico* dice come ordinare (notate la parola istruzioni che è in **grassetto**)
6. Una lista ordinata da le tre istruzioni; la parola "truffe" è un link all'indirizzo di posta elettronica truffe@ggo.com



Politecnico di Milano - Anno Accademico 2001-2002 - Informatica B
Recupero – 25 Febbraio 2002 – **Elaborato A**

NOME COGNOME	
Matricola	

```
<html>
<head>
  <title>Primo Esercizio di Recupero</title>
</head>
<body>
<h1> Q.Q.O. Quadri Quasi Originali</h1>
<center><table >
<tr>
<th>Nome del quadro</th>
<th>Quadro</th>
<th>Prezzo</th>      </tr>

<tr>
<td><a href="http://www.qgo.com/bimbe.html">Bimba</a></td>
<td>
<center><img SRC="Bimbe.gif" height=80 width=80></center>
</td>
<td>&euro; 1.350</td>      </tr>

<tr>
<td><a href="http://www.qgo.com/cavalli.html">Cavalli</a></td>
<td>
<center><img SRC="Cavalli.gif" height=80 width=162></center> </td>
<td>&euro; 1.990</td>      </tr>
</table></center> <p>
<i><blink>Per ordinare seguite le <b>istruzioni</b></blink> </i>
<ol>
<li> Scegliete il quadro</li>
<li> Scrivete al nostro servizio <a
href="mailto:truffe@qgo.com">truffe</a></li>
<li> Pagate in biglietti di piccolo taglio</li>
</ol>

</body>
</html>
```