

Politecnico di Milano - Anno Accademico 2001-2002 - Informatica B
2° Recupero – 20 Settembre 2002 – Elaborato A

NOME COGNOME	
Matricola	

REC2 /Punti	I / 2 punti	II / 1 punti	III / 2 punti	IV /1 punto	V /4 punti	REC1
Tot= / 10						

REC2 I – Calcolatore – Tradurre il semplice programma da Assembler in C e determinare il risultato numerico. (si consiglia di **NON stravolgere** l'impostazione) (2 punti)

	<i>Assembler</i>	<i>Linguaggio C</i>
EXIT	EQU 2000	Void main()
POS	DW ?	#define LUNG 3
STAMPANTE	EQU 2	
LUNG	EQU 3	int POS;
TAB	DW 1	int TAB[LUNG]={1,2,3};
	DW 2	
	DW 3	
RISULTATO	DW ?	Int RISULTATO ;
INIZIO	LDA #0	{ POS = 0 ;
	STA POS	
	STA RISULTATO	RISULTATO = 0 ;
CICLO	LDA POS	while (POS != LUNG)
	SUB #LUNG	{
	JZ PRINT	
	LDA RISULTATO	RISULTATO = RISULTATO +
	LDI POS	TAB[POS];
	ADD TAB(I)	
	STA RISULTATO	
	LDA POS	POS=POS+1;
	ADD #1	
	STA POS	
	JMP CICLO	} /* end while
PRINT	LDA RISULTATO	Scrivi RISULTATO ;
	OUT STAMPANTE	
	JMP EXIT	} /* end main
Il risultato stampato è		6

Politecnico di Milano - Anno Accademico 2001-2002 - Informatica B
2° Recupero – 20 Settembre 2002 – Elaborato A

NOME COGNOME	
Matricola	

REC2 – II - Cosa stampa il seguente programma? (1 punto)

```
void main(void)
{
int a, b, c, d;
a = 2;
b = 3;
d = 4;
b -= 3;
a=b;
if (a == 17) c = ++a + ++b;
else c = a++ + d;      d -=
3;
printf("c = %d, d = %d\n", c,
d);
}
```

OUTPUT :

c = 4, d = 1
perché a++ viene eseguito DOPO!!

REC2 - III – Per ognuna delle espressioni logiche riportate nelle righe della tabella, assumendo che $v1 = 10$, $v2 = 20$ e $v3 = 30$, indicare nella colonna *a* se l'espressione è vera o falsa (scrivere V o F). Indicare inoltre, nella colonna *b*, se l'espressione sarebbe vera per qualsiasi valore delle tre variabili, e nella colonna *c* se l'espressione sarebbe falsa per qualsiasi valore delle tre variabili (scrivere SI o NO). (2 punti)

espressione	a vera o falsa?	b sempre vera?	c sempre falsa?
$v1 < 10 \parallel v1 > 10$	F	NO	NO
$v1 > v2 \ \&\& \ v2 > v1$	F	NO	SI
$(v1 > v2) \ \&\& \ (v1 + 10 \leq 20)$	F	NO	NO
$v1 > v2 \parallel v2 \geq v1$	V	SI	NO
$(v1 == v2) \parallel (v1 != v2) \ \&\& \ (v1 < v2 \parallel v2 < v1)$	V	SI	NO

Politecnico di Milano - Anno Accademico 2001-2002 - Informatica B
2° Recupero – 20 Settembre 2002 – Elaborato A

NOME COGNOME

Matricola

REC2 - IV – Riscrivi le istruzioni eseguibili del seguente frammento di codice usando solo cicli while e rispondere alla domanda sottostante (totale 1 punto)

```
int i , j, n, s;

s = 0;
for (i = 0; i < 10; i++) {
    n = 0;
    for (j = 0; j <= i; j++)
    {
        n = n + 1;
    }
    s = s + n;
}
```

Soluzione

```
s = 0;
i = 0;
while (i < 10) {
    n = 0;
    j = 0;
    while (j <= i) {
        n = n + 1;
        j = j + 1;
    }
    s = s + n;
    i = i + 1;
}
```

(b) Quanto vale la variabile *s* al termine dell'esecuzione del codice? **R. 55**

REC2 - V – programma (4 punti)

Scrivere un programma C che legge tutti i caratteri del file di testo `dati.txt`, supposto già esistente, e li scrive sul file `uscita.txt` trasformando in minuscolo i caratteri alfabetici maiuscoli; i caratteri già minuscoli e quelli non alfabetici vanno ricopiati così come sono sul file `uscita.txt`. Al termine dell'esecuzione il programma visualizza su standard output il numero di caratteri trasformati. La soluzione dovrà seguire le seguenti linee:

1. opportuna gestione dei file interessati; (0,5 punti);
2. trasformazione da carattere maiuscolo in minuscolo (2 punti);
3. conteggio dei caratteri (1 punto);
4. stampa del numero dei caratteri trasformati (0,5 punti).

Politecnico di Milano - Anno Accademico 2001-2002 - Informatica B
2° Recupero – 20 Settembre 2002 – Elaborato A

NOME COGNOME	
Matricola	

/* nella valutazione i punti 2 e 3 comprendono la corretta gestione del ciclo. Ho anche omesso di dire che il punto 3 va svolto contemporaneamente al 2. Non mi sono sentito di dare queste indicazioni, ma se vogliamo facilitarli si possono evidenziare. Sono stato vago sul primo punto che comprende anche la chiusura del file*/

```
#include <stdio.h>
int trasforma(void)
{ FILE *in, *out;
  int cont;
  char c ;

  if((in=fopen("dati.txt","r")) == NULL)
    {printf("errore apertura dati.txt\n");return(-1);}
  if((out=fopen("uscita.txt","w")) == NULL)
    {printf("errore apertura uscita.txt\n");return(-2);}
  cont=0;

  while(fscanf(in,"%c",&c) != EOF) {
    if((c>='A') && (c<='Z'))
      {c=c-('a'-'A'); cont=cont+1;}
    fprintf(out,"%c",c);
  }
  fclose(in);fclose(out);
  return(cont);
}

main()
{ int a;
  a=trasforma();
  printf("%d\n",a);
  return(0);
}
```