



POLITECNICO DI BARI
Corso di Laurea in Ingegneria
Informatica e dell'Automazione
(docente: V.Bevilacqua)

I PROVA DI ESONERO
DI FONDAMENTI DI INFORMATICA II del 4/06/2005

Esercizio 1: Si completi la seguente tabella delle pagine fino all'istante $t=5$, e si indichino possibili dimensioni degli spazi di indirizzamento logico/fisico, nell'ipotesi che si sappia che all'istante $t=1$, si sia rilocato l'indirizzo logico 129, nell'indirizzo fisico 1, che all'istante $t=4$ venga rilocato l'indirizzo logico 15, che all'istante $t=5$, dopo una chiamata dell'algoritmo LRU, venga rilocato l'indirizzo logico 113. Si indichino in ciascuno degli istanti compresi fra $t=1$ e $t=5$, quali indirizzi logici sono stati rilocati ed in quali indirizzi fisici.

N° pagina logica	Bit di Presenza	Istante di rilocazione	N° pagina fisica
0000			
0001			
0010			
0011	1	3	
0100			
0101			
0110			
0111			
1000	1	1	
1001			
1010			
1011			
1100			
1101	1	2	
1110			
1111			

Esercizio 2: si scrivano due segmenti di codice in linguaggio C dai quali si evidenzi che avendo una pagina di 32 byte ed un page frame di 4 pagine, si hanno un numero di page fault notevolmente diversi e se ne discuta il loro valore.

Esercizio 3: si dimensiona un circuito decodificatore attraverso il quale sia possibile correggere la trasmissione di un messaggio avente un pay-load di 4 bit. Si fornisca un esempio numerico di tale correzione.

Esercizio 4: si fornisca un esempio, dal quale sia evidente che utilizzando il metodo delle mappe di Karnaugh si ottiene un risparmio di esattamente 17 porte AOI, rispetto alla forma canonica di una funzione booleana di 4 variabili.

Esercizio 5: si scriva il prototipo di una funzione che implementa una pila per mezzo di una lista.

Esercizio 6: Si completi e corregga il seguente codice in modo tale che l'eseguibile scriva il file uscita.txt.

Si modifichi in seguito, il sorgente corretto, utilizzando un puntatore a funzione, in modo tale che la funzione di stampa su file stampi solo i numeri dispari.

```
#include <stdio.h>
#include <conio.h>
void alloca(... , ...);
void setta (int *, int);
void stampa_su_file(int, int);

void main()
{
    int n;
    printf("quanti elementi =");
    scanf("%d",&n);
    alloca(... , ...);
    setta(&v,n);
    stampa_su_file(v,n);
    free(*v);
    stampa_su_file(v,n);
}

void alloca(int *v, int n)
{v=malloc(n*sizeof(int));}

void setta(int *v, int n)
{
for(i=0;i<n;i++)
v[i]=0;
}

void stampa_su_file(int *v, int n)
{
for(i=0;i<n;i++)
printf(" .... ", i, ...,v[i]);
}
```

Il seguente file uscita.txt è nella directory di C che ha il tuo cognome

v[0] che si trova ai byte numero -29 -30 -31 -32 vale =0
v[1] che si trova ai byte numero -25 -26 -27 -28 vale =1
v[2] che si trova ai byte numero -21 -22 -23 -24 vale =2
v[3] che si trova ai byte numero -17 -18 -19 -20 vale =3
v[4] che si trova ai byte numero -13 -14 -15 -16 vale =4

Esercizio 7 (OBBLIGATORIO): fornire una sezione del codice (prototipi, main e funzione) di un programma che chiama una funzione che riceve per indirizzo una lista bidirezionale e scrive in un file binario tutti gli elementi che si trovano fra due numeri pari, prima di eliminare gli stessi dalla lista e di ripassare il controllo al main che dovrà visualizzare la lista modificata.

Esercizio 8: scrivere un codice in C nel quale sia evidente i vantaggi e gli svantaggi di allocare matrici staticamente e dinamicamente.

Esercizio 9: fornire un esempio di codice in C che lavora sui grafi, allocando dinamicamente la matrice delle adiacenze.

Esercizio 10: fornire un esempio di codice in C dal quale sia evidente l'utilità della funzione fseek, ftell e della macro SEEK_CUR.