

Prendete il vostro numero di matricola. Eliminate tutti gli “0”, e chiamate  $N$  il numero risultante. Esempio se il numero di matricola e’ 0000800470, allora  $N = 847$

Nel seguito  $N_i$  e’ la cifra numero  $i$  di  $N$ , contando a partire da 0, modulo la lunghezza di  $N$ . Nell’esempio sopra:

- $N_0 = 8$
- $N_1 = 4$
- $N_2 = 7$
- $N_3 = 8$
- $N_4 = 4$

e cosi’ di seguito. Quindi per esempio  $N_2N_3$  e’ il numero 78 (e NON la moltiplicazione  $7 \times 8$ ).

Negli esercizi sotto: non eseguite calcoli o ragionamenti astratti in termini di questi  $N_i$ , ma sostituite subito ai vari  $N_i$  il valore concreto dato dal vostro numero matricola.

---

**Ad inizio foglio indicate chiaramente il vostro numeri di matricola e quasi sono nel vostro caso i numeri  $N_0, N_1, N_2, N_3$**  Motivare le risposte date. Non e’ necessario dare risposte molto lunghe, ma e’ importante rispondere in modo motivato ed esauriente alle domande poste (in altre parole, molto meglio una frase in piu’ che una in meno). Ricordare che lo scan finale deve contenere il libretto universitario di riconoscimento

---

**E’ necessario svolgere gli esercizi 1 e 2**

---

**Esercizio 1** Come in esercizi a lezione, date 2 lettere H,K, usiamo la notazione  $\{HK\}$  per indicare le due sequenze possibili HK e KH. Quindi per esempio “stampare  $L\{HK\}M$ ” significa essere in grado di stampare le 2 stringhe LHKM e LKHM (e nessun altra). Considerate i seguenti processi:

P1	P2
<code>print(A)</code>	<code>print(D)</code>
<code>print(B)</code>	<code>print(E)</code>
<code>print(C)</code>	<code>print(F)</code>

Se  $N_2$  e  $N_3$  sono entrambe pari, seguite la opzione (a) sotto; se sono entrambe dispari, seguite la opzione (b) sotto; se  $N_2$  e’ pari e  $N_3$  e’ dispari, seguite la opzione (c) sotto; se  $N_2$  e’ dispari e  $N_3$  e’ pari, seguite la opzione (d) sotto.

Agite, se necessario, sul codice, inserendo opportune operazioni su semaforo, in modo che l’output di questa esecuzione concorrente siano esattamente le stringhe

- (a)  $\{AD\}BECF$
- (b)  $A\{BD\}ECF$
- (c)  $AD\{BE\}CF$
- (d)  $ADE\{BF\}C$

**Risposta**(Sketch) Nelle soluzione sotto, tutti i semafori inizializzati a 0

(a) {AD}BECF

P1	P2
print(A)	print(D)
P(S)	V(S); P(T)
print(B)	print(E)
V(T); P(U)	V(U); P(Z)
print(C)	print(F)
V(Z)	

(b) A{BD}ECF

P1	P2
print(A)	P(T)
V(T);	print(D)
print(B)	P(S)
V(S); P(U)	print(E)
print(C)	V(U) ; P(Z)
V(Z)	print(F)

(c) AD{BE}CF

P1	P2
print(A)	P(T)
V(T); P(S)	print(D)
print(B)	V(S)
P(U)	print(E)
print(C)	V(U); P(Z)
V(Z)	print(F)

(d) ADE{BF}C

P1	P2
print(A)	P(T)
V(T); P(S)	print(D)
print(B)	print(E)
P(U)	V(S)
print(C)	print(F)
	V(U)

**Esercizio 2** In un sistema operativo che adotta uno scheduling di tipo Round Robin con quanto di tempo 3, quattro processi arrivano al tempo indicato e consumano la quantita' di CPU indicata nella tabella sottostante:

Processo	T. di arrivo	Burst time
A	0	$N_2 + 5$
B	4	$N_3 + 3$
C	7	$N_1$
D	10	$N_2$

1. Riportate il diagramma di Gantt che mostra cio' che succedera' in esecuzione.
2. Qual e' il waiting time di B?

**Risposta**(Sketch) Svolgimento, assumendo  $N_2 = 7, N_3 = 6, N_1 = 8$  Quindi abbiamo

Processo	T. di arrivo	Burst time
A	0	12
B	4	9
C	7	8
D	10	7

Riporto Gantt ed evoluzione pila:

[0] A [3] A [6] B [9] A [12] C [15] B [18] D [21] A [24]  
 C [27] B [30] D [33] C [35] D [36]

A  
 A  
 BA  
 ACB  
 CBDA  
 BDAC  
 DACB  
 ACBD  
 CBD  
 BDC  
 DC  
 CD  
 D

waiting time B:  $30 - 4 - 9 = 19$

**Esercizio 3** Se  $N_2 \leq N_3$  andate al punto (a); se  $N_2 > N_3$  andate al punto (b).

- (a) In un SO la tabella delle pagine puo' contenere al massimo  $2^{N_1 N_2}$  entry. Nell'indirizzo fisico, la parte offset e' fatta da  $N_1 N_3$  bit.
- (b) In un SO la tabella delle pagine puo' contenere al massimo  $2^{N_1 N_3}$  entry. Nell'indirizzo fisico, la parte offset e' fatta da  $N_1 N_2$  bit.

In entrambi i casi (a) e (b) le domande a cui rispondere sono le seguenti:

1. Quando e' grande un frame, in byte?
2. Quanti sono i bit dell'indirizzo logico?
3. Quanti frame deve avere questo sistema per poter dire con certezza che il sistema deve implementare la memoria virtuale?

4. Quanti byte dovrebbero contenere ciascuna entry della tabella delle pagine per dover adottare un sistema di paginazione a piu' livelli (cioe' con la tabella delle pagine paginata su piu' livelli)?

**Risposta**(Sketch) Caso (a)

1.  $2^{N_1 N_3}$
2.  $N_1 N_2 + N_1 N_3$
3. Meno di  $2^{N_1 N_2}$  frame
4. Dovrebbe valere  $2^{N_1 N_2} \times x > 2^{N_1 N_3}$ , quindi piu' di  $2^{N_1 N_3 - N_1 N_2}$

Caso (b)

1.  $2^{N_1 N_2}$
2.  $N_1 N_3 + N_1 N_2$
3. Meno di  $2^{N_1 N_3}$  frame
4. Dovrebbe valere  $2^{N_1 N_3} \times x > 2^{N_1 N_2}$ , quindi piu' di  $2^{N_1 N_3 - N_1 N_2}$

**Esercizio 4** Sia  $M$  il minimo tra i numeri 5 e  $N_2$ .

- (a) Usare le proprieta' dell'aritmetica modulo per calcolare in modo semplice

$$(N_2 \times 7 + M)^{N_3 + 5} \bmod 7$$

(quindi la base dell'esponenziazione e' il numero  $(N_2 \times 7 + M)$ , l'esponente e' il numero  $N_3 + 5$ ).

- (b) Riportare la distinzione tra crittografia a chiave simmetrica e a chiave pubblica. Quali sono i principali vantaggi della seconda?

**Risposta**(Sketch) Svolgimento, assumendo  $N_2 = 5, N_3 = 6$

Intanto, la domanda e' la stessa di chiedere  $N^{N_3 + 5} \bmod 7$ . Assumo  $M = 5$ . Quindi interessa  $5^{11} \bmod 7$ .

Abbiamo:  $5^2 \bmod 7 = 4$ . Quindi  $5^4 \bmod 7 = 4^2 \bmod 7 = 2$ .

Quindi  $5^8 \bmod 7 = 2^2 \bmod 7 = 4$ .

Siccome  $11 = 8 + 2 + 1$ , diventa  $(4 + 4 + 5) \bmod 7 = 6$ .

Per la parte (b), vedere note corso

**Esercizio 5** Riportate lo pseudocodice che descrive l'implementazione dell'operazione di P(S) su un semaforo. Che problemi di atomicita' (cioe' di interferenze possibili) presenta? (A lezione si sono viste 2 possibili implementazioni di semaforo, sto facendo riferimento alla prima, dove il semaforo e' anche chiamato "spinlock")

**Risposta**(Sketch) vedere note corso.

**Esercizio 6** 1. Qual'e' lo svantaggio principale del metodo di "File allocation" chiamato "Linked list" ?

2. Quali sono, sempre in questo metodo, i possibili pro e contro che emergono al variare delle dimensioni dei blocchi?

**Risposta**(Sketch)

1. Accesso diretto ai record logici inefficiente. Nel caso peggiore, un file system deve accedere tutti i blocchi di un file (caso di singly-linked list) per trovare i dati richiesti, o meta' (caso di doubly-linked list).
2. Un blocco grande riduce il numero di operazioni IO richieste per ritrovare un certo record, ma al costo di maggiore frammentazione interna; riducendo la dimensione e' il contrario

**Esercizio 7** 1. In una politica di CPU scheduling a priorit , qual'  il problema principale?

2. Perch  nei sistemi operativi moderni la politica Round Robin   quella pi  diffusa per gestire processi utente?

**Risposta**(Sketch) vedere note corso.