

# Compito (tempo a disposizione: 2 ore e 30 minuti)

- **DOMANDA 1** (40 minuti) **Compilare la seguente function.**

**N.B. scansione con puntatore e valore soglia**

*\$a0, \$a1, \$a2, \$a3*

```
int esempio (int A[],int B[],int n,  
            int k)
```

```
{
```

```
    int i; int app;
```

*\$s0*

```
    i = 0;
```

```
    while i != n {
```

```
        app = f(A[i]);
```

*\$a0*

```
        if app < 0
```

```
            B[i] = k;
```

```
        else B[i] = A[i];
```

```
        i = i + 1;
```

```
    }
```

```
    return app;
```

```
}
```

---

- **DOMANDA 2** (20 minuti)

Con riferimento alla implementazione *multiciclo* della CPU della macchina MIPS senza pipelining, definire il concetto di *microprogramma* e delineare uno schema per l'unità di controllo.

- **DOMANDA 3** (20 minuti)

Presentare schematicamente ma senza ambiguità le trasformazioni subite dall'indirizzo generato dalla CPU in un sistema di memoria comprendente sia cache che memoria virtuale (con TLB), *evidenziando il ruolo e le dimensioni dei vari campi dell'indirizzo*

---

■ **DOMANDA 4** (30 minuti)

Fornire una formulazione dettagliata del tempo di CPU relativo all'utente in una macchina con un livello di cache, **discutendo in particolare gli aspetti architetturali che incidono sul valore delle varie componenti e le interdipendenze tra tali componenti.**

■ **DOMANDA 5** (20 minuti)

Con riferimento alla *pipeline con schedulazione dinamica*:

1. discutere esaurientemente il ruolo e le attività delle unità di consegna specificando perchè sono necessarie
2. mostrare le differenze tra il PowerPC e il PentiumPro **relativamente al trattamento dei bit prelevati dalla cache istruzioni**, motivando tali differenze e delineandone le conseguenze

## Domanda 6 (20 minuti)

- **Valutare il numero di cicli di clock necessari a completare l'esecuzione di questo programma nel caso di pipeline semplice (si ipotizzi che il ciclo si esegue 100 volte).**
  - 1. Si supponga di essere nel caso ideale di assenza di fallimenti in cache
  - 2. Si adotti la tecnica basata sulla predizione di fallimento per la gestione delle criticità sul controllo
  - 3. Si adotti la tecnica di propagazione per la risoluzione delle criticità sui dati
  - 4. Si supponga di non poter effettuare il riordinamento del codice per criticità di tipo carica-e-usa
  - Discutere la realizzazione hardware della tecnica di propagazione.

```
add $t0,$t1,$t3
add $t1,$zero,$a0
loop: beq $t0,$t1,exit
      lw  $t3,0($t0)
      sw  $t3,0($t0)
      add $t5,$s1,$t3
      addi $t0,$t0,4
      j  loop
exit: sw $s1,0($s4)
```

- ***N.B. Evidenziare tutti gli anticipi e gli stalli necessari, specificando il numero di volte per cui si ripetono***