

Programmazione Funzionale e Parallela (A.A. 2015-2016)

Corso di Laurea in Ingegneria Informatica e Automatica
Sapienza Università di Roma

B

Esame del 10/02/2015 – Durata 1h 30' (solo esonerati)

Inserire nome, cognome e matricola nel file `studente.txt`.

Esercizio 1 (Filtri grafici mediante OpenCL)

Lo scopo dell'esercizio è quello di scrivere un modulo C basato su OpenCL che, data in input un'immagine a 256 toni di grigio di dimensione $w \times h$, crei una nuova immagine di dimensioni $w \times 2h$, come nell'esempio sotto.



(a) Immagine originale



(b) Immagine mirror

Si completi nel file `mirror/mirror.c` la funzione `mirror` con il seguente prototipo:

```
void mirror(unsigned char* in, int w, int h,  
            unsigned char** out, int* ow, int* oh,  
            clut_device* dev, double* td)
```

dove:

- `in`: puntatore a un buffer di dimensione $w \times h \times \text{sizeof}(\text{unsigned char})$ byte che contiene l'immagine di input in formato row-major¹;
- `w`: larghezza di `in` in pixel (numero di colonne della matrice di pixel);
- `h`: altezza di `in` in pixel (numero di righe della matrice di pixel);
- `out`: puntatore a puntatore a buffer di dimensione $w \times 2h \times \text{sizeof}(\text{unsigned char})$ byte che deve contenere l'immagine di output in formato row-major; **il buffer deve essere allocato nella funzione `mirror`**;
- `ow`: puntatore a `int` in cui scrivere la larghezza di `out` in pixel;
- `oh`: puntatore a `int` in cui scrivere l'altezza di `out` in pixel.

Per compilare usare il comando `make`. Per effettuare un test usare `make test`. Verrà prodotta l'immagine di output `colosseo-mirror.pgm`.

¹ Cioè con le righe disposte consecutivamente in memoria.

Esercizio 2 (Confronto prefissi di array mediante vettorizzazione)

Lo scopo dell'esercizio è scrivere una funzione che confronta se un array è prefisso di un altro array usando vettorizzazione SSE.

Si completi nel file `vpref/vpref.c` la funzione `vpref` con il seguente prototipo:

```
int vpref(const char* a, const char* b, int na, int nb);
```

dove:

- `a`: puntatore al primo vettore;
- `b`: puntatore al secondo vettore;
- `na`: numero di elementi del primo vettore;
- `nb`: numero di elementi del secondo vettore.

La funzione deve restituire 1 se `a` è un prefisso di `b` oppure `b` è un prefisso di `a`, e 0 altrimenti.

Suggerimento. Usare le seguenti funzioni SSE:

- `__m128i _mm_cmpeq_epi8(__m128i a, __m128i b)` [SSE 2]: restituisce un oggetto vettoriale i cui bit sono posti tutti a 1 se e solo se `a` è uguale a `b`.
- `int _mm_test_all_ones(__m128i a)` [SSE 4.1]: restituisce 1 se tutti i bit di `a` sono 1, e 0 altrimenti.

Compilare da riga di comando usando `gcc` con il main di prova fornito e generare un eseguibile chiamato `vpref`.