Esercizio 1: Tradurre il seguente codice C in assembly LEGv8.

```c
void swap(long long int* x, long long int* y)
{
	long long int t;
	t = *y;
	*y = *x;
	*x = t;
}

long long int median3(long long int a, long long int b, long long int c)
{
	if (a > c)
		swap(&a, &c);

	if (a > b)
		swap(&a, &b);

	if (b > c)
		swap(&b, &c);

	return b;
}
```

Nota: Gli argomenti vanno passati a `swap` tramite i registri x0, x1 e a `median3` tramite i registri x0, x1, x2 e il risultato va restituito tramite x0. In `median3` assumere che `swap` possa modificare qualunque “registro non preservato tra chiamate”.

Esercizio 2: Tradurre il seguente codice C in assembly LEGv8.

```c
void FIR_filter(long long int h[], long long int N, long long int x[],
		long long int LEN, long long int y[])
{
	long long int n;
	long long int i;
	long long int S;
	for (n = N; n < LEN; n++) {
		S = 0;
		for (i = 0; i <= N; i++)
			S += h[i] * x[n - i];
		y[n - N] = S;
	}
}

// Suggerimento: i parametri vengono passati in X0, ..., X4
// Specificare con commenti l'uso fatto dei registri, e.g. X9 -> n, e le operazioni principali, e.g. // S = 0
```

Esercizio 3: Tradurre il seguente codice C in assembly LEGv8.

```
void autocorrelation(long long int x[], long long int y[], long long int LEN)
{
	long long int n;
	long long int m;
	long long int A;
	for (n = 0; n < LEN; n++) {
		A = 0;
		for (m = 0; m < LEN - n; m++)
			A += x[m] * x[m + n];
		y[n] = A;
	}
}

// Suggerimento: i parametri vengono passati in X0, X1, X2
// Specificare con commenti l'uso fatto dei registri, e.g. X9 -> n, e le operazioni
principali, e.g. // S = 0
```

Esercizio 4: Tradurre il seguente codice C in assembly LEGv8.

```
void updateX(long long int x, long long int X[], long long int N);

long long int FIR(long long int x, long long int h[], long long int X[],
		  long long int N)
{
	long long int i;
	long long int S;
	updateX(x, X, N);
	S = 0;
	for (i = 0; i < N; i++)
		S += h[i] * X[i];
	return S;
}

// Suggerimento: i parametri di updateX vengono passati in X0, X1, X2;
// i parametri di FIR vengono passati in X0, X1, X2, X3 e il risultato in X0.
// Specificare con commenti l'uso fatto dei registri, e.g. X9 -> i, e le operazioni
principali, e.g. // S = 0
```

Esercizio 5: Tradurre il seguente codice C in assembly LEGv8.

```
void IIR2_filter(long long int x[], long long int c[], long long int LEN,
                 long long int y[])
{
      long long int i;
      long long int A;
      long long int z[2];
      z[0] = 0;
      z[1] = 0;
      for (i = 0; i < LEN; i++) {
            A = x[i] - z[0] * c[1] - z[1] * c[2];
            y[i] = c[3] * A + c[4] * z[0] + c[5] * z[1];
            z[1] = z[0];
            z[0] = A;
      }
}

// Suggerimento: i parametri di IIR2_filter vengono passati in X0, X1, X2, x3
//                e z[] va salvato nello stack.
// Specificare con commenti l'uso fatto dei registri, e.g. X9 -> i, e le operazioni
principali, e.g. // Z[1] = 0
```