

# Architectures des Ordinateurs Avancé: Sujets d'études de cas - 2020

Emmanuel Oseret – Kevin Camus  
(emmanuel.oseret@uvsq.fr) – (kevin.camus@uvsq.fr)

## Sujet 1

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n, double x[n],
               double y[n], double z[n][n]) {
    unsigned i, j;

    for (j=0; j<n; j++)
        for (i=0; i<n; i++)
            x[i] += z[i][j] / y[i];
}
```

Compilateur et options de référence: gcc -O2.

## Sujet 2

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n, float a[n], float b[n],
               float c[n], float d[20]) {
    unsigned k, i;

    // all elements in b are assumed positive
    for (k=0; k<20; k++) {
        for (i=0; i<n; i++) {
            if (b[i] >= 0.0 && b[i] < 1.0) {
                a[i] += exp (b[i] + d[k]) / c[i];
            } else if (b[i] >= 1.0) {
                a[i] += (b[i] * d[k]) / c[i];
            }
        }
    }
}
```

Compilateur et options de référence: gcc -O3.

## Sujet 3

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n, float a[n], float b[n],
               float c[n]) {
    unsigned i, j;

    for (i=0; i<n; i++) {
        if (i < n/2)
            c[i] = a[i] + b[i];
        else
            c[i] = a[i] * b[i];
    }

    for (i=1; i<n; i++)
        c[i] *= 2;
}
```

Compilateur et options de référence: gcc -O2 ou icc -O2.

## Sujet 4

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n, double a[n], unsigned ind[n],
               double b[n], double c[n][n]) {
    unsigned i, j;

    for (j=0; j<n; j++)
        for (i=0; i<n; i++)
            c[i][j] = a[ind[j]] / b[i];
}
```

Compilateur et options de référence: gcc -O3.

## Sujet 5

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n, float m[n][n], float x[n],
               float y[16]) {
    unsigned i, j, k;

    for (i=0; i<n; i++)
        x[i] = 0;

    for (k=0; k<16; k++)
        for (j=0; j<n; j++)
            for (i=0; i<n; i++)
                x[i] += m[i][j] * y[k];
}
```

Compilateur et options de référence: gcc -O3.

## Sujet 6

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n, float x[n][n],
               float x0[n][n], double c) {
    unsigned i, j;

    for (j=2; j<n-2; j++)
        for (i=2; i<n-2; i++)
            x[i][j] = ((x[i-2][j+0] +
                        x[i-1][j+0] -
                        x[i+1][j+0] +
                        x[i+2][j+0] -
                        x[i+0][j-2] +
                        x[i+0][j-1] -
                        x[i+0][j+1] +
                        x[i+0][j+2])
                       + x0[i][j]) / c;
}
```

Compilateur et options de référence: gcc -O2 ou icc -O2.

## Sujet 7

Étudier et optimiser la fonction C suivante:

```
typedef struct {
    float x;
    float y;
    float z;
} elem_t;

elem_t baseline (unsigned n, elem_t a[n][n]) {
    unsigned i, j;
    elem_t s = { 1.0f, 1.0f, 1.0f };

    for (j=0; j<n; j++)
        for (i=0; i<n; i++) {
            s.x += a[i][j].x;
            s.y += a[i][j].y;
            s.z += a[i][j].z;
        }

    return s;
}
```

Compilateur et options de référence: gcc -O2.

## Sujet 8

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n, float a[n],
               float b[n], float c[n]) {
    unsigned i, j;

    for (j=0; j<n; j++)
        for (i=0; i<n; i++) {
            c[i] -= b[n - 1 - i];
            if (i < j)
                c[i] += a[j];
        }
}
```

Compilateur et options de référence: gcc -O2.



## Sujet 9

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n, float a[n][n],
               float b[n], float c[n]) {
    unsigned i, j;

    for (j=0; j<n; j++)
        for (i=0; i<n; i++)
            a[i][j] = exp (b[i] + c[j]);
}
```

Compilateur et options de référence: gcc -O2 ou gcc -O3.

## Sujet 10

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n, float a[n], float b[n],
               float x) {
    unsigned i;

    for (i=0; i<n; i++) {
        if ((i < n/3) && (a[i] > x))
            b[i] = a[i];
        else if (i < n/3)
            b[i] = x;
        else
            b[i] = a[i] + x;
    }

    for (i=0; i<n; i++) {
        if (b[i] < 0.0) b[i] = 0.0;
    }
}
```

Compilateur et options de référence: gcc -O2.

## Sujet 11

Étudier et optimiser la fonction C suivante:

```
void baseline (unsigned n,
               float a[n][n],
               float b[n][n],
               float c[n][n]) {
    unsigned i, j;

    for (i=0; i<n; i++) {
        for (j=0; j<n-1; j++)
            a[j][i] = b[i][j] - 1;
        for (j=0; j<n-1; j++)
            c[i][j] = a[j+1][i] * b[i][j+1];
    }
}
```

Compilateur et options de référence: gcc -O2.

## Sujet 12

Étudier et optimiser la fonction C suivante:

```
float baseline (unsigned n, double a[n][n]) {
    unsigned i, j;
    float s = 0.0;

    for (j=0; j<n; j++)
        for (i=0; i<n; i++)
            s += a[i][j];

    return s;
}
```

Compilateur et options de référence: gcc -O2 ou gcc -O3.