

Architectures des Ordinateurs Avancé: Sujets d'études de cas - 2021

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

Compilateur et options de référence: `gcc -O2`.

Sujet 1

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

    for (i=0; i<n; i++) {
        if (i < n/3)
            c[i] = a[i] * b[i];
        else
            c[i] = a[i] - b[i];
    }

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

Sujet 2

```
#include <stdlib.h> // malloc, free

void baseline (unsigned n, float a[n], float b[n]) {
    unsigned i;
    double s = 0.0;
    float *tmp = malloc (n * sizeof *tmp);

    for (i=0; i<n; i++) {
        tmp[i] = i + b[i];
        s += tmp[i];
    }

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

    free (tmp);
}
```

Sujet 3

```
#include <math.h> // exp

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

    for (i=0; i<n; i++) {
        if (i%4 == 0) {
            a[i] = exp (1.0f + b[i]) / c[i];
        } else if (i%4 == 1) {
            a[i] = (1.0f + b[i]) / c[i];
        } else {
            a[i] = (1.0f - b[i]) / c[i];
        }
    }
}
```

Sujet 4

```
#include <math.h> // sqrt

//Remark:  $i1 \neq i2 \Leftrightarrow perm[i1] \neq perm[i2]$ 
void baseline (unsigned n, float a[n],
               unsigned perm[n], float b[n][n]) {
    unsigned i, j;

    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            b[i][j] = 0.0;

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

Sujet 5

```
void baseline (unsigned n, float m[n][n], float v1[n],
               float v2[20]) {
    unsigned i, j, k;

    for (i=0; i<n; i++)
        v1[i] = 0;

    for (k=0; k<20; k++)
        for (j=0; j<n; j++)
            for (i=0; i<n; i++)
                v1[i] += m[i][j] / v2[k];
}
```

Sujet 6

```
typedef struct {
    float re;
    float im;
} complex_t;

void baseline (unsigned n, complex_t a[n],
               complex_t b[n]) {
    unsigned i;

    for (i=0; i<n; i++) {
        a[i].re = a[i].re + b[i].im;
        b[i].im = b[i].im * b[i].im;
    }
}
```

Sujet 7

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

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


Sujet 8

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

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

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

Sujet 9

```
typedef struct {
    float t;
    double p;
    int v;
} elem_t;

elem_t baseline (unsigned n, elem_t a[n][n]) {
    unsigned i, j;
    elem_t s = { 0.0f, 0.0, 0 };

    for (j=0; j<n; j++)
        for (i=0; i<n; i++) {
            s.t += a[i][j].t;
            s.p += a[i][j].p;
            s.v += a[i][j].v;
        }

    return s;
}
```

Sujet 10

```
#include <math.h> // log

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] = log (b[i] * c[j]);
}
```

Sujet 11

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

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

    return s;
}
```

Sujet 12

```
#include <math.h> // sqrt

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

Sujet 13

```
void baseline (unsigned n, const float a[n],
               const float b[n], float c[n][n],
               int offset, double radius) {
    unsigned i, j;

    for (i=0; i<n; i++) {
        for (j=0; j<n; j++) {
            if (offset+j < 0 || offset+j >= n) continue;

            c[i][offset+j] = 0.0;
            if (a[offset+j] < radius) {
                c[i][offset+j] = a[offset+j] / b[i];
            }
        }
    }
}
```