

# TP 2 : tableaux et polynomes

Programmation en C (LC4)

Semaine du 5 février 2007

## ► Exercice 1

```
int separe(int i, int j, unsigned int t[], int x) {
    unsigned int tmp;
    int k = i;
    int y = 1<<x;
    for (; i <= j; i++) {
        if ((t[i]&y) == 0) {
            tmp=t[i];
            t[i]=t[k];
            t[k]=tmp;
            k++;
        }
    }
    return k;
}
```

## ► Exercice 2

```
void tri_rec(int i, int j, unsigned int t[],int x) {
    int k;
    if (x<0) return;
    k=separe(i,j,t,x);
    if (k>i+1) tri_rec(i,k-1,t,x-1);
    if (k<j) tri_rec(k,j,t,x-1);
}

void tri(int n,unsigned int t[]) {
    tri_rec(0,n-1,t,31);
}
```

## ► Exercice 3

```
double value(double P[], int d, double x) {
    int i;
    double y,z;
    for (i = 0, y = 1, z = 0; i <= d; i++, y *= x)
        z += y*P[i];
    return z;
}
```

```

/* une autre version utilisant l'algorithme de Horner (bien plus efficace) */
double evaluate(double P[], int d, double x){
    int i;
    double z;
    for(i = d-1 , z = P[d]; i >= 0 ; i--)
        z = z*x + P[i];
    return z;
}

```

► **Exercice 4**

```

double racine(double P[], int d, double a, double b, double precision) {
    double c;
    int sgnc, sgna = evaluate(P,d,a)>0;
    while ( (b-a) > precision) {
        c = (a+b)/2;
        sgnc = evaluate(P,d,c)>0;
        if (sgnc == sgna)
            a=c;
        else
            b=c;
    }
    return a;
}

```

► **Exercice 5**

```

int fact(int n){
    if (n == 0)
        return 1;
    else
        return(n * fact(n-1) );
}

double binomial(int n, int p) {
    return fact(n) / (fact(p) * fact(n-p));
}

void decalage(double P[],int d,double x) {
    int i, j;
    double y;
    for (i = 1; i <= d; i++)
        for (j = i-1, y = x; j >= 0; j--, y *= x)
            P[j] += pow(P[i],i-j) * binomial(i,j) * y;
}

```

► **Exercice 6**

```

int ordonnee(double y, double y0, double precision) {
    int j = floor( (y-y0) / precision);
}

```

```

    return j;
}

void trace(double P[], int d, double precision, double x, double y,
           int largeur, int hauteur, int image[]) {
    int i, j, k, l;
    j = ordonnee(evaluate(P, d, x), y, precision);
    if (j >= 0 && j < hauteur)
        image[j*largeur]=0;
    for (i = 1; i < largeur; i++, x += precision, j = k) {
        k = ordonnee(evaluate(P, d, x), y, precision);
        if (j < 0 || j >= hauteur || k < 0 || k >= hauteur) continue;
        if (j < k) {
            for(l = j; l < (j+k)/2; l++)
                image[i-1 + largeur*l]=0;
            for(; l <= k; l++)
                image[i + largeur*l]=0;
        }
        else {
            for(l = k; l < (j+k)/2; l++)
                image[i + largeur*l]=0;
            for(; l <= j; l++)
                image[i-1 + largeur*l]=0;
        }
    }
}
}
}

```