Exercice 23 :

1. $P_1(p) = 1$ ssi p calcule une fonction totale.

$\exists p_0$ tq $P_1(p_0) = 0$

```
int p0 (int x) {while (1), return 0;}
```

$\exists p_1$ tq $P_1(p_1) = 1$

```
int p1 (int x) { while (1), return x;}
```

P_1 n'est pas trivial, donc le thm de Rice s'applique.

P_1 n'est pas calculable.

Supposons P_1 calculable par procP1,

```
int gamma1(int x) {
    if (procP1(gamma1)) {
        while(1);
    }
    else return x;
}
```

$P_1(\gamma_1) = 1 \Rightarrow \forall x$ gamma x boucle

$\Rightarrow$ gamma 1 ne calcule pas une fonction totale $\Rightarrow P_1(\gamma_1) = 0$ contradiction

$P_1(\gamma_1) = 0 \Rightarrow \forall x \quad \text{gamma1}(x) = x$

$\Rightarrow$ gamma 1 calcule une fonction totale

$\Rightarrow P_1(\gamma_1) = 1$ contradiction

Donc l'hypothèse de départ est fausse

et il n'existe pas proc P_1 qui calcule P_1.

2. $P_2(p) = 1$ ssi p calcule une

fonction totale.

$\exists p_0$ tq $P_2(p_0) = 0$

```
int p0 (int x) { while (1), return x % 3 }
```

$\exists p_1$ tq $P_2(p_1) = 1$

```
int p1 (int x) { while (1), return x; }
```

P_2 n'est pas trivial, donc le thm de Rice s'applique.

P_2 n'est pas calculable.

Supposons P_2 calculable par procP_2,

```
int gamma2(int x) {
    if (procP2(gamma2)) {
        return x % 3
    }
    else return x;
}
```

3. $P_3(p) = 1$ ssi p calcule une fonction croissante.

$\exists p_0$ tq $P_3(p_0) = 0$

int p0 (int x) { while (1), return x % 3 }

$\exists p_1$ tq $P_3(p_1) = 1$

int p1 (int x) { while (1), return x; }

P_3 n'est pas trivial, donc le thm de Rice s'applique.

P_3 n'est pas calculable.

Supposons P_3 calculable par proc P_3,

```
int gamma3 (int x) {
        if (procP3( gamma3)) {
                return x % 3
        }
        else return x;
}
```

4. $P_4(p) = 1$ ssi p calcule une fonction à valeur bornée.

$\exists p_0$ tq $P_4(p_0) = 0$

```
int p0 (int x) { return x; }
```

$\exists p_1$ tq $P_3(p_1) = 1$

```
int p1 (int x) { return x; }
```

P_4 n'est pas trivial , donc le thm

de Rice s'applique.

P_4 n'est pas calculable.

Supposons P_4 calculable par procP_4,

$\boxed{D}$ à terminer

Excercice 24:

A est un ensemble récursivement énumérable

et $P : A \to Bool$ est un programme total

$B = \{ a \in A \mid P(a) = vrai \}$

Mq B est récursivement énumérable

Soit f la f^n totale calculable qui énumère

les éléments de A.

Pour tout n faire

Si $P(f(n))$ alors

afficher $f(n)$

Exercice 29 :

Soit A et B décidable :

f_A = fonction caractéristique de A.

f_B = fonction caractéristique de B.

$f_C\ A \cup B$ (int x) return $f_A(x) + f_B(x)$
$- f_A(x) * f_B(x)$

$f_C\ A \cap B$ (int x) return $f_A(x) * f_B(x)$

1. $\overline{A}$:

```
int fcompA(int x) {
        return 1 - fA(x);
}
```

4.2. Soient A et B récursivement énumérable, f qui énumère A, g qui énumère B avec f et g calculable

$A \cup B$: pour n de 0 à $+\infty$ faire :

afficher $(f(n))$

afficher $(g(n))$

Rq : on affiche 2 fois certains éléments.

Donc $A \cup B$ est récursivement énumérable.

4.3. Pour tout couple (i,j) faire :

Si $f(i)=g(j)$ alors :

afficher $f(i)$

Exercice 25 :

f calculable partielle, E récursivement énumérable

$E = \{ val(f) \}$

Pour tout couple (n, t) faire :

Si $h(p, n, t)$ alors :

afficher $p(n)$

Exercice 27 :

$E = \{ f(0), f(1), \dots, f(i), \dots \}$ avec

f calculable, totale, croissante.

Mq E est décidable :

```
int fcE(int x) {
        for(int n=0; ; n++) {
                if(f(n)==x) return 1;
                if(f(n)>x) return 0;
        }
}
```

$$\left.\begin{array}{l} x \neq f(i),\ \forall i \leqslant n \\ f(n) > x \end{array}\right\} \Rightarrow x \notin E$$

si f est croissante strictement.

2. $F = \{ \underset{\overset{\shortparallel}{g(0)}}{f(0)},\ \underset{\overset{\shortparallel}{g(1)}}{f(i)}$

$\forall j\ ,\ j < i \qquad f(j) \leqslant f(0)\ ,$

$f(i) > f(0)$

F croissante, F est décidable.

Exercice 30 :

1.

$E = \{ x \mid \exists y$ tel que $(x,y) \in A \}$ est récursivement énumérable

On sait faire car A est décidable

$(x,y) \in A$.

Pour tout couple (x,y) faire :

Si $(x,y) \in A$ alors :

afficher (x)

Donc E est récursivement énumérable.

2. Réciproquement E est récursivement énumérable tq $E = \{ f(0), f(1), f(2), \ldots, f(i), \ldots \}$ avec f totale et calculable

$G = \{ (n, f(n)) \} = \{ (0, f(0)), (1, f(1)), \ldots \}$

G est décidable

```
int fcG(int x, int y) {
    return y == f(x)
```