

# CIL 2008 • Epreuve de Demi-finale

## a) Solutions modèles en Delphi-Pascal

### Problème I - Parfaits

30 points

Pour écrire un programme rapide, il faut avoir compris qu'on n'a pas besoin de tester tous les diviseurs jusqu'au nombre N, mais qu'il suffit de tester jusqu'à la racine carrée. On trouve les diviseurs supérieurs à la racine aisément en calculant la division de N par les diviseurs inférieurs à la racine.

```
somme := somme + i + n div i
```

Exemple:

```
div 32 =    1 et donc 32
           2 et donc 16
           4 et donc  8
```

Pour le nombre 10000 par exemple, on obtient ainsi une boucle allant jusqu'à 100, au lieu d'aller jusqu'à 10000!

```
program PARFAITS;

var F : text;
    N,T : integer;

function PARF (N : integer) : boolean;
var SOMME,I : integer;
begin
    SOMME := 1;
    for I := 2 to round(sqrt(N)) do
        if N mod I = 0
        then SOMME := SOMME + I + N div I;
    PARF := SOMME = N
end;

begin
    assign(F, 'PARFAITS.TXT');
    rewrite(F);
    N := 2;
    T := 0;
    while T < 4 do
        begin
            if PARF(N)
            then begin
                write(F,N, ' ');
                T := T + 1
            end;
            N := N + 1
        end;
    close(F)
end.
```

La méthode utilisée pour trouver rapidement le nombre est celle de la recherche dichotomique. A chaque tour, la plage des nombres restants est divisée en deux, éliminant ainsi la moitié des nombres. Vu que la plage initiale va de 1 à 1000, on considère la moitié chaque fois à partir de 1000:

1000, 500, 250, 125, 62, 31, 15, 7, 3, 1

Après 10 fois, il ne reste qu'un seul nombre.

$$2^{10} = 1024 \approx 1000 \quad \text{ou bien} \quad \log_2 1000 < 10$$

Un programme intelligent propose donc au premier tour le nombre 500, milieu entre 1 et 1000. Selon la réponse de l'utilisateur, il reste soit la moitié inférieure, de 1 à 499, soit la moitié supérieure, de 501 à 1000. Une moitié de la plage est éliminée! La proposition suivante sera donc soit 250, soit 750!

```

program DEVIN;

var LIMINF, LIMSUP, CHOIX, N : integer;
    REP : string;
    TROUVE : boolean;

begin
    writeln('Cherchez un nombre entre 1 et 1000 !');
    LIMINF := 1;
    LIMSUP := 1000;
    TROUVE := false;
    N := 0;
    repeat
        CHOIX := (LIMINF + LIMSUP) div 2;
        writeln('Mon choix : ', CHOIX);
        N := N + 1;
        write('Votre réponse : ');
        readln(REP);
        if REP = 'T'
        then TROUVE := true
        else if REP = 'G'
            then LIMSUP := CHOIX - 1
            else LIMINF := CHOIX + 1
    until TROUVE;
    writeln('J''ai trouvé votre nombre en ', N, ' fois !');
    readln
end.

```

Pour montrer la méthode, nous avons laissé les affichages temporaires dans le code, entre accolades pour commentaires. Si on enlève les accolades, on peut suivre l'évolution du programme à l'écran. Pour la programmation pratique, ces affichages sont fortement recommandés. N'oubliez quand même pas de les enlever, ou de les marquer comme commentaires, avant de remettre vos programmes!

```
program CODE;  
  
var F : text;  
    STR : string;  
    DIM : integer;  
    A : array [1..16,1..16] of string;  
    I,J,K : integer;  
  
begin  
    assign(F, 'CODEIN.TXT');  
    reset(F);  
    readln(F, STR);  
    close(F);  
    {writeln(STR);}  
    DIM := round(sqrt(length(STR)));  
    {writeln(DIM);}  
    K := 1;  
    for I := 1 to DIM do  
        for J := 1 to DIM do  
            begin  
                if K <= length(STR)  
                then A[I,J] := copy(STR, K, 1)  
                else A[I,J] := ' ';  
                K := K + 1  
            end;  
        {for I := 1 to DIM do  
            begin  
                for J := 1 to DIM do  
                    write(A[I,J], ' ');  
                writeln  
            end;}  
    STR := '';  
    for I := 1 to DIM do  
        for J := 1 to DIM do  
            STR := STR + A[J,I];  
        {writeln(STR);}  
    assign(F, 'CODEOUT.TXT');  
    rewrite(F);  
    writeln(F, STR);  
    close(F);  
    {readln}  
end.
```

CIL 2008 • Questionnaire de l'Epreuve de Sélection Préliminaire  
b) Solutions modèles en C

**Problème I - Parfaits**

**30 points**

```
#include <math.h>
#include <stdio.h>

int parf(int n)
{
    int somme, i;

    somme=1;
    for (i = 2; i <= (int)(sqrt(n)+0.5); i++)
        if (n%i == 0)
            somme += i+n/i;
    return(somme == n);
}

void main(void)
{
    int n, t;
    FILE *f;

    f = fopen ("PARFAITS.TXT", "w");
    n=2;
    t=0;
    while(t<4)
    {
        if(parf(n))
        {
            fprintf(f,"%d ", n);
            t++;
        }
        n++;
    }
    fprintf(f, "\n");
    fclose(f);
}
```

```
#include <stdio.h>
#include <string.h>

#ifdef TRUE
    #define TRUE 1
#endif
#ifdef FALSE
    #define FALSE 0
#endif

void main(void)
{
    int  liminf, limsup, choix, n, trouve;
    char rep[2];

    printf("Cherchez un nombre entre 1 et 1000 !\n");
    liminf = 1;
    limsup = 1000;
    trouve = FALSE;
    n      = 0;
    do
    {
        choix = (liminf + limsup) / 2;
        printf("Mon choix : %d\n", choix);
        n++;
        printf("Votre réponse : ");
        scanf("%s", &rep[0]);
        if (strncmp(rep, "T", 1) == 0)
            trouve = TRUE;
        else
            if (strncmp(rep, "G", 1) == 0)
                limsup = choix - 1;
            else
                liminf = choix + 1;
    }
    while(!trouve);
    printf("J'ai trouvé votre nombre en %d fois !\n", n);
}
```

```
#include <utility.h>
#include <stdio.h>
#include <string.h>
#include <math.h>

void main(void)
{
    FILE *f;
    char matrix[16][16],
        readstr[255],
        writestr[255];
    int readstrlen,
        matrixsize,
        col, lin,
        strpos;

    f = fopen ("CODEIN.TXT", "r");
    fgets(readstr, sizeof(readstr)/sizeof(char), f);
    fclose(f);

    // filter out linefeed
    if (readstr[strlen(readstr)-1]==10)
        readstr[strlen(readstr)-1] = 0;
    readstrlen=strlen(readstr);

    // calc number of rows/columns of matrix
    if (sqrt(readstrlen)==(int)sqrt(readstrlen))
        matrixsize = (int)sqrt(readstrlen);
    else
        matrixsize = (int)sqrt(readstrlen)+1;

    // initialize matrix (col,line)
    strpos=0;
    for (col=0; col < matrixsize; col++)
        for (lin=0; lin < matrixsize; lin++)
            if (strpos < readstrlen)
                matrix[col][lin]=readstr[strpos++];
            else
                matrix[col][lin]=' ';

    // transpose contents (line, col) to string
    strpos=0;
    for (lin=0; lin < matrixsize; lin++)
        for (col=0; col < matrixsize; col++)
            writestr[strpos++]=matrix[col][lin];
    writestr[strpos]=0x00;

    // filter out trailing spaces
    strpos=strlen(writestr)-1;
    while(writestr[strpos]==' ')
        strpos--;
    writestr[strpos+1]=0x00;
```

```
// write to file
f = fopen ("CODEOUT.TXT", "w");
fputs(writestr,f);
fclose(f);
}
```