Uitwerking tentamen Programmeermethoden 8 januari 2007

OPGAVE 1

a.
```
int draaiom (int x) {
  int res = 0;
  while ( x != 0 ) {
    res = 10 * res + x % 10;
    x = x / 10;
  }//while
  return res;
}//draaiom
```

b.
```
bool kleiner (int x, int y) {
  x = draaiom (x);
  y = draaiom (y);
  while ( x % 10 == y % 10 ) {
    x = x / 10;
    y = y / 10;
    if ( y == 0 ) return false; else if ( x == 0 ) return true;
  }//while
  return ( x % 10 < y % 10 );
}//kleiner
```

c.
```
int kleinste (int A[ ], int n) {
  int i, kl = 0; bool eerste = true;
  for ( i = 1; i < n; i++ )
    if ( kleiner (A[i],A[kl]) ) { kl = i; eerste = true; }
    else if ( A[i] == A[kl] && eerste ) { kl = i; eerste = false; }
  return kl;
}//kleinste
```

d.
```
void sorteer (int A[ ], int n) {  // bubblesort
  int i, j, temp;
  for ( i = 1; i < n; i++ )
    for ( j = 0; j < n-i; j++ )
      if ( kleiner (A[j+1],A[j]) ) {
        temp = A[j]; A[j] = A[j+1]; A[j+1] = temp;
      }//if
}//sorteer
```

OPGAVE 2

a. Globale variabelen gelden in het gehele programma, en worden helemaal bovenin aangemaakt. Locale variabelen gelden (tijdelijk) alleen in de functie waarin ze aangemaakt zijn.
Variabelen kunnen call by value en call by reference worden meegegeven aan een functie. Bij call by value gaat alleen de waarde van de parameter naar de functie, alwaar een locale variabele deze waarde opvangt, en er met deze locale variabele wordt verder gerekend. De oorspronkelijk variabele behoudt zijn waarde. Bij call by reference (&) gaat als het ware de variabele zelf naar de functie, en kan dan ook blijvend veranderd worden. Eigenlijk wordt het adres (de reference) doorgegeven.
Formeel: in functieheading, bijvoorbeeld x, y in int f (int x, bool y) {
Actueel: bij aanroep, bijvoorbeeld r en y in z = f (r,y);

b. Jan -5 en 1 Jan -5 en 2 Jan -5 en 3 Jan -5 en 4 Jan -5 en 5
Wout 6 en 5 42, 1 en 5

c. Jan -5 en 1 Jan -1 en -4 Wout 0 en -4
5, 0 en -4

d. Jan -670 en 670 Wout 671 en 670
2007, 1 en 5

e. Dan moet boven janpeter nog een prototype van wouter worden gezet:
```
 int wouter (int i, int grens);
```
Let er op dat de recursie een keer afbreekt = stopt, het basisgeval dus.

OPGAVE 3

a.
```
bool slaan (int puz[m][n], int i, int j, int r, int s) {
  return ( 0 <= i && i < m && 0 <= j && j < n
           && 0 <= r && r < m && 0 <= s && s < n && ( i != r || j != s )
           && ( i == r || j == s || i+j == r+s || i-j == r-s );
}//slaan
```

b.
```
int tel (int puz[m][n], int i, int j, int nr) {
  int telze = 0, r, s;
  for ( r = 0; r < m; r++ ) for ( s = 0; s < n; s++ )
    if ( slaan (puz,i,j,r,s) && puz[r][s] == 0 ) {
      telze++; puz[r][s] = nr; }//if
```

```
    return telze;
  }//tel
c. int kan (int puz[m][n], int i, int j, int r, int s) {
    int nr = 1, telze = tel (puz,i,j,-nr);
    while ( telze != 0 && puz[r][s] == 0 ) {
      telze = 0;
      for ( i = 0; i < m; i++ ) for ( j = 0; j < n; j++ )
        if ( puz[i][j] == -nr )
          telze += tel (puz,i,j,-nr-1);
      nr++;
    }//while
    return -puz[r][s];
  }//kan

OPGAVE 4
a. void voegtoe (info* & ingang, char let, int get) {
    info* nieuw = new info;
    nieuw->letter = let; nieuw->getal = get;
    nieuw->volgg = ingang; ingang = nieuw;
  }//voegtoe
b. void verwijder (info* & ingang) {
    info* weg = ingang;
    if ( ingang != NULL
         && ( ingang->letter != 'D' || ingang->getal != 66 ) ) {
      ingang = ingang->volgg; delete weg;
    }//if
  }//verwijder
c. void verwissel (info* & ingang) {
    info* p;
    if ( ingang != NULL & ingang->volgg != NULL
         && ingang->getal > ingang->volgg->getal ) {
      p = ingang->volgg;
      ingang->volgg = p->volgg;
      p->volgg = ingang;
      ingang = p;
    }//if
  }//verwissel
d. Bij a, b en c moet er een & bij: de pointer ingang gaat veranderen!
  (Bij b soms niet, overigens.)
e. void voegtoe (info* & ingang, char let, int get) {
    info* p = ingang; info* q = NULL; info* nieuw;
    while ( p != NULL ) {  // zoek grootste kleinere
      if ( p->letter < let && ( q == NULL || p->letter > q->letter ) )
        q = p;
      p = p->volgg;
    }//while
    nieuw = new info;
    nieuw->letter = let; nieuw->getal = get;
    nieuw->volgg = ingang; ingang = nieuw;
    nieuw->volgl = q->volgl; q->volgl = nieuw;
  }//voegtoe
```