

Uitwerkingen Tentamen Programmeermethoden maandag 2 april 2007

OPGAVE 1

```
a.void bubblesort (int A[ ], int n) {
    int ronde, i, temp;
    for ( ronde = 1; ronde < n; ronde++ )
        for ( i = 0; i < n - ronde; i++ )
            if ( A[i] > A[i+1] ) {
                temp = A[i]; A[i] = A[i+1]; A[i+1] = temp; }//if
    }//bubblesort
b.bool saai (int x) {
    int cijfer = x % 10;
    while ( x != 0 ) {
        if ( x % 10 != cijfer ) return false;
        x = x / 10; }//while
    return true;
    }//saai
c.void saaivooraan (int A[ ], int n) {
    int i = 0, j = n-1, temp;
    while ( i < j ) {
        if ( saai (A[i]) ) i++;
        else if ( ! saai (A[j]) ) j--;
        else {
            temp = A[i]; A[i] = A[j]; A[j] = temp; i++; j--; }//if
        }//while
    }//saaivooraan
d.Zie a, maar nu met test
    if ( A[i] > A[i+1] && saai (A[i]) == saai (A[i+1]) )
```

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.3, 9 en 4 <CR> 4, 16 en 13 <CR> 5, 25 en 29 <CR> 2, 4 en 29
 4, 16 en 9 <CR> 5, 25 en 25 <CR> 3, 4 en 54
 5, 25 en 16 <CR> 4, 4 en 70
 2, 4 en 70 (met wat regelovergangen <CR> erbij)

c.3, 9 en 4 <CR> 4, 16 en 13 <CR> 5, 25 en 29 <CR> 5, 4 en 29
 6, 4 en 29

d.3, 9 en 4 <CR> 3, 2 en 4 <CR> 2, 4 en 4

e.De som van de kwadraten van a tot en met b (en dus 0 als a > b).

OPGAVE 3

```
a.double gem (kost[n][n], int i) {
    int som = 0, aantal = 0, j;
    for ( j = 0; j < n; j++ )
        if ( kost[i][j] > 0 ) { som += kost[i][j]; aantal++; }//if
    return ( (double) som ) / aantal; // aanname: aantal > 0
    }//gem
b.int verschil (int kost[n][n]) {
    int i, j, gr = 0, ver;
    for ( i = 0; i < n; i++ )
        for ( j = 0; j < n; j++ ) {
            if ( kost[i][j] > kost[j][i] ) ver = kost[i][j] - kost[j][i];
            else ver = kost[j][i] - kost[i][j];
            if ( ver > gr ) gr = ver; }//for
    return gr;
    }//verschil
c.int hoeveel (int kost[n][n], int i) {
    bool D[n]; int j, k, telvorige = 0, telze = 1;
    for ( j = 0; j < n; j++ ) D[j] = false;
    D[i] = true;
    while ( telze > telvorige ) {
        for ( j = 0; j < n; j++ )
```

```

        if ( D[j] ) for ( k = 0; k < n; k++ )
            if ( kost[j][k] > 0 ) D[k] = true;
        telvorige = telze;
        for ( j = 0; j < n; j++ ) if ( D[j] ) telze++;
    }//while
    return telze;
}//hoeveel

```

OPGAVE 4

```

a.void voegtoe (mens* & ingang, char nm, int lt, bool mv) {
    mens* nieuw = new mens;
    nieuw->naam = nm; nieuw->leeftijd = lt;
    if ( mv ) { nieuw->volg1 = ingang; nieuw->volg2 = NULL; }//if
    else {
        nieuw->volg1 = NULL; nieuw->volg2 = ingang;
        if ( ingang == NULL ) nieuw->leeftijd = -lt; }//else
    ingang = nieuw;
}//voegtoe
b.void verwijderman (mens* & ingang) {
    mens* weg = ingang;
    if ( ingang != NULL &&
        ( ingang->leeftijd < 0 || ingang->volg2 != NULL ) ) {
        ingang = ingang->volg2; delete weg; }//if
}//verwijderman
c.void verwissel (mens* ingang) {
    int temp; mens* twee;
    if ( ingang != NULL &&
        ( ingang->volg1 != NULL || ingang->volg2 != NULL ) ) {
        if (ingang->volg1 != NULL) twee = ingang->volg1;
        else twee = ingang->volg2;
        temp = ingang->leeftijd;
        ingang->leeftijd = twee->leeftijd;
        twee->leeftijd = temp;
        if ( ingang->leeftijd < 0 ) {
            ingang->leeftijd = -ingang->leeftijd;
            twee->leeftijd = - twee->leeftijd; }//if
    }//if
}//verwissel
d.Bij a, b moet er een & bij: de pointers moeten kunnen veranderen
Bij b verandert ingang niet altijd overigens.
Bij c hoeft geen &, de ingangspointer verandert niet, alleen de inhoud.
e.int vrouwman (mens* ingang) {
    mens* loper = ingang; int tel = 0;
    while ( loper != NULL ) {
        if ( loper->volg1 != NULL ) { // dit is een vrouw
            if ( loper->volg1->volg2 != NULL || loper->volg1->leeftijd < 0 )
                tel++; // gevolgd door een man
            loper = loper->volg1;
        }//if
        else loper = loper->volg2;
    }//while
    return tel;
}//vrouwman

```