

Uitwerking Tentamen Programmeermethoden 1 april 2005

OPGAVE 1

```

a. int aantal (int A[ ], int n) {
    int i, teller = 0;
    for ( i = 0; i < n-1; i++ )
        if ( ( A[i] >= 0 && A[i+1] < 0 ) || ( A[i] < 0 && A[i+1] >= 0 ) )
            teller ++;
    return teller;
} //aantal

b. void wissel (int A[ ], int i, int j) {
    int temp = A[i]; A[i] = A[j]; A[j] = temp;
} //wissel

c. void bubblesort (int A[ ], int n) {
    int ronde, i;
    for ( ronde = 1; ronde < n; ronde++ )
        for ( i = 0; i < n-ronde; i++ )
            if ( A[i] < A[i+1] )
                wissel (A, i, i+1);
} //bubblesort

d. void negpos (int A[ ], int n) {
    int i = 0, j = n-1;
    while ( i < j )
        if ( A[i] >= 0 && A[j] < 0 ) {
            wissel (A, i, j); i++; j--; } //if
        else if ( A[i] < 0 ) i++;
        else j--; // dus als A[j] >= 0
} //negpos

```

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. 3,7,4,2 3,7,4,2 2,4,3,1 1,4,3
- c. 3,7,4,2 3,10,4,2 2,10,3,1 1,10,2
- d. Oneindige loop in de functie een, omdat de bovengrens van de loop steeds wordt opgehoogd
- e. Bij een (y, z) : z wordt $z+y$, y blijft onveranderd
En ander (y, z) : z wordt met 1 afgehoogd, y wordt opgehoogd met $y * \text{ceiling}(x/2)$ (waarbij ceiling naar boven afrondt)

OPGAVE 3

```

a. int tel (int eennul[ ][n]) {
    int teller = 0, i, j;
    for ( i = 0; i < n; i++ )
        for ( j = 0; j < n; j++ )
            if ( eennul[i][j] == 0 ) teller++;
    return teller;
} //tel

b. void som (int eennul[ ][n], int i, int j, int & rijsom, int & kolomsom) {
    int k, rijsom = 0; kolomsom = 0;
    for ( k = 0; k < n; k++ ) {
        rijsom += eennul[i][k]; kolomsom += eennul[k][j]; } //for
} //som

c. bool kruis (int eennul[ ][n]) {

```

```

int i, j;
for ( i = 0; i < n; i++ )
    for ( j = 0; j < n; j++ )
        if ( ( i == j || i == n-1-j ) && eennul[i][j] == 1 ) return false;
        else if ( ( i != j && i != n-1-j ) eennul[i][j] == 0 ) return false;
return true;
} //tel
d. int nultrap (int eennul[ ][n]) {
    int tel = 0, i = 0, j = 0;
    while ( j+1 < n ) { // dan ook i+1 < n
        j++;
        if ( eennul[i][j] != 0 ) return tel;
        tel++;
        i++;
        if ( eennul[i][j] != 0 ) return tel;
        tel++;
    } //while
    return tel;
} //nultrap

```

OPGAVE 4

- a. void verander (resultaat* begin, int code, int punt) {
 if (begin->volgende->vakcode == code)
 begin->volgende->cijfer = punt;
 else begin->volgende->volgende->cijfer = punt;
} //verander
- b. void voegtoe (resultaat* & begin, int code, int year, int punt) {
 resultaat* nieuw = new resultaat;
 nieuw->vorige = NULL; nieuw->vakcode = code;
 nieuw->jaar = year; nieuw->cijfer = punt;
 begin->vorige = nieuw; // gegeven: begin niet NULL
 nieuw->volgende = begin; begin = nieuw;
} //voegtoe
- c. void doeweg (resultaat* & begin, resultaat* & eind) {
 resultaat* weg = eind; // gegeven: niet NULL
 if (begin == eind) begin = NULL;
 eind = eind->vorige;
 if (eind != NULL) eind->volgende = NULL;
 delete weg;
} //doeweg
- d. void gooiweg (resultaat* & begin, resultaat* & eind, int year) {
 while (eind != NULL && eind->jaar < year)
 doeweg (begin, eind);
} //gooiweg
- e. Bij a hoeft er geen & bij (de pointer begin verandert niet),
 bij b, c en d moet het wel (bij c verandert soms de pointer begin niet,
 bij d hoeven ze ook niet altijd te veranderen).