

Uitwerking Tentamen Programmeermethoden maandag 5 januari 2004

- ```

1.a. void sorteer (char A[], int n) {
 int i, j;
 char temp;
 for (i = 1; i < n; i++)
 for (j = 0; j < n-i; j++)
 if (A[j] < A[j+1]) {
 temp = A[j]; A[j] = A[j+1]; A[j+1] = temp; } //if
 } //sorteer
b. int getal (char A[], int n) {
 int res = 0, i;
 for (i = 0; i < n; i++)
 res = 10 * res + (A[i] - '0');
 return res;
 } //getal
c. void som (char a, char b, char & c, bool & over) {
 c = (a - '0' + b - '0') % 10 + '0';
 over = (a - '0' + b - '0' < 10);
 } //som
d. bool totaalSom (char A[], char B[], char C[], int n) {
 int i;
 bool over = true;
 for (i = n-1; i >= 0; i--)
 if (over)
 som (A[i],B[i],C[i],over);
 else
 som ((char)(A[i]+1),B[i],C[i],over);
 // of nog een keer '1' erbij optellen, en opletten met over's
 return over;
 } //totaalSom
2.a. formeel: in functieheading, bijvoorbeeld int f (int x, bool y) { ...
actueel: bij aanroep, bijvoorbeeld r en y in z = f (r,y);
call by value: waarde wordt doorgegeven aan locale kopie; actuele
parameter verandert niet
call by reference (met & erbij): variabele zelf wordt doorgegeven, en
kan dus ook veranderen
globaal: kan overal gebruikt worden; bovenaan programma aangemaakt
locaal: plaatselijk in functie geldig
b. 4 18 6 16 1 4
c. 4 18 6 16 18 6
d. 4 17 5 16 17 5 Wordt de oude of de gewijzigde n opgeteld?
 4 16 5 15 16 5
e. Met prototype int paul (int n, int m); erboven mag het
 Het is dan recursie: de functies roepen zichzelf (indirect) aan.
3.a. bool tweemaal (int cijfers[][n], int i, int j) {
 int k;
 bool okee = true;
 for (k = 0; k < n; k++)
 if (A[i][k] != 2 * A[j][k])
 okee = false; // of meteen return false;
 return okee;
 } //tweemaal
b. int fraai (int cijfers[][n], int m) {
 int i, j, tel = 0, cnt = 1;
 for (i = 0; i < m; i++)
 for (j = 0; j < n; j++) {
 if (cijfers[i][j] != cnt) // of keihard n*i+j+1
 tel++;
 cnt++;
 } //for
 return tel;
 }

```

```

 }//fraai
c. void schuifdoor (int cijfers[][n], int j, int m) {
 int i, temp = cijfers[m-1][j];
 for (i = m-1; i > 0; i--)
 cijfers[i][j] = cijfers[i-1][j];
 cijfers[0][j] = temp;
}//schuifdoor
d. int hoevaak (int cijfers[][n], int m) {
 int i = 0, j = 0, tel = 0;
 bool rechts = true;
 while (i < m && j < n) {
 if (rechts)
 j += cijfers[i][j];
 else
 i += cijfers[i][j];
 tel++;
 rechts = !rechts;
 }//while
 return tel-1;
}//hoevaak

4.a. void verwijder (film* marathon) {
 film* hulp = marathon->volgende;
 marathon->volgende = hulp->volgende;
 delete hulp;
}//verwijder
b. void toevoegen (film* & marathon, char titel, int duur) {
 film* hulp = new film;
 hulp->naam = titel;
 hulp->lengte = duur;
 hulp->volgende = marathon;
 marathon = hulp;
}//toevoegen
c. void verbeter (film* marathon, char titel, int duur) {
 if (marathon->naam == titel)
 marathon->lengte = duur;
 else
 marathon->volgende->lengte = duur;
}//verbeter
d. Bij b moet er een & bij: de ingangspointer zal gaan veranderen.
 Bij a en c maakt het niet uit, de ingangspointer verandert toch niet.
e. int hoeveel (film* marathon) {
 int lang = 0, tel = 0, som = 0;
 film* hulp = marathon;
 while (hulp != NULL) {
 tel++;
 som += hulp->lengte;
 hulp = hulp->volgende;
 }//while
 hulp = marathon;
 while (hulp != NULL) {
 if (hulp->lengte > (double) som / tel)
 lang++;
 hulp = hulp->volgende;
 }//while
 return lang;
}//hoelang

```