

2 Invoer en Arrays

Oefening 12

```
#include <stdio.h>
#define AANTAL 3 /* voor testdoeleinden zet je dit klein genoeg! */
int main(){
    int getal;
    int som = 0;
    int aantal = 1;
    printf("Geef een positief geheel getal: ");
    scanf("%d",&getal);
    while(aantal < AANTAL && getal >= 0){
        som += getal;
        aantal++;
        printf("Geef een positief geheel getal: ");
        scanf("%d",&getal);
    }
    if(getal < 0){
        printf("U gaf helaas geen %d positieve gehele getallen op.\n",AANTAL);
    }
    else{
        som += getal; /* verwerken van het laatste getal, dat ook positief was */
        printf("Dank, u gaf %d positieve gehele getallen op.\n",AANTAL);
    }
    printf("De som van de positieve getallen die u opgaf is %d.",som);
    return 0;
}
```

Oefening 13

```
#include <stdio.h>
/* in functie/procedure kan sizeof niet achterhalen hoe groot de array is*/
void schrijf_even_posities(const char tabel[], int aantal){
    printf("IN PROCEDURE #elementen =%i\n", sizeof(tabel)/sizeof(int));
    int i;
    for(i=0; i<aantal; i+=2){
        printf("%c",tabel[i]);
    }
}

int main(){
    char letters [] =
        {'p','o','r','e','o','i','f','o','i','e','c','i','i',':','a','-','t','('};
    int lengte, i;

    lengte = sizeof(letters)/sizeof(char);
    printf("lengte van array met letters is %i\n",lengte);
    for(i=0; i<lengte; i+=2){
        printf("%c",letters[i]);
    }

    printf("\nLetters uitschrijven via procedure\n");
    schrijf_even_posities(letters,lengte);
    return 0;
}
```

Oefening 14

```

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

#define EPSILON 0.000001; /* een klein getal (doubles vergelijk je nooit exact !)*

int index_van(const double[], int , double);
int index_van_geordend(const double[], int ,double);

int main(){
    double rij[] = {8.8,4.4,2.2,6.6,0.0,10.0};
    double x;
    int lengte = sizeof(rij)/sizeof(double);
    int index;

    printf("Geef een reeel getal op ");
    while(scanf("%lf",&x)==0){
        while(getchar()!='\n'){
            printf("."); /*visualiseert wat hier gebeurt*/
        }
        printf("Geef een reeel getal op ");
    }
    printf("\nJe gaf het getal %.1f op.",x);

    index = index_van(rij,lengte,x);
    if(index==-1){
        printf("\n%.1f werd niet gevonden in de niet-geordende array.",x);
    }
    else{
        printf("\nIn de niet-geordende array staat %.1f op plaats %d.\n",x,index);
    }

    return 0;
}

/*      Merk op: Je MOET stoppen zodra je vond wat gezocht werd.
        Je MAG NIET de hele array doorlopen als je al vond wat je zocht!! */

int index_van(const double array[], int aantal, double gezocht){
    int i = 0;
    while(i<aantal && fabs(gezocht-array[i]) > EPSILON){
        i++;
    }
    return (i<aantal ? i : -1);
}

int index_van_geordend(const double array[], int n, double x){
    int i=0;
    while (i<n && array[i] < x ){
        i++;
    }
    return ( i<n && fabs(array[i]-x) < EPSILON ? i : -1);
}

```

Oefening 15

```
#include <stdio.h>

void schrijf(const char rij[],int aantal);
void schuif_links(char rij[], int lengte);

int main(){

    char rij[] = {'s','a','p','a','p','p','e','l'};
    int lengte = sizeof(rij)/sizeof(char);
    int i;

    schrijf(rij,lengte);

    for(i = 0; i < 3; i++){
        schuif_links(rij,lengte);
        schrijf(rij,lengte);
    }

    return 0;
}

void schrijf(const char rij[],int aantal){
    int i;
    for(i=0; i<aantal; i++){
        printf("%c",rij[i]);
    }
    printf("\n");
}

void schuif_links(char rij[], int lengte){
    char reserve = rij[0];
    int i;
    for(i=1; i<lengte; i++){
        rij[i-1] = rij[i];
    }
    rij[lengte-1] = reserve;
}
```

Oefening 16

```
#include <stdio.h>
#define MIN 100
#define MAX 120
#define BEREIK MAX-MIN
#define AANTAL MAX-MIN

int main(){

    int getal;
    int aanwezigheid[BEREIK+1] = {0}; /*alle elementen zullen hierdoor 0 zijn*/
    int i;

    srand(time(NULL)); /* anders krijg je altijd diezelfde getallen! */
```

```
/* let op! als je - ter controle - hier in de lus
   ook wil uitschrijven welk getal er effectief gegenereerd werd,
   zorg er dan voor dat je slechts EEN MAAL rand() oproept!!! */

for(i = 0; i < AANTAL; i++){
    getal = MIN + rand()%(BEREIK+1);
    aanwezigheid[getal - MIN] = 1;
    printf("%d ",getal);
}

printf("\nKwamen niet voor:\n");
for(i = 0; i < BEREIK+1; i++){
    if (aanwezigheid[i] == 0){
        printf("%d ",(i+MIN));
    }
}

return 0;
}
```

Oefening 17

```
#include <stdio.h>

void print_lijn(char c, int lengte);

void teken_horizontaal(const int* frequenties);
void teken_vertikaal(const int* frequenties);

int is_kleine_letter(char c);
int is_grote_letter(char c);

void lees_input(int* frequenties);

int main(){
    int frequenties[26];
    lees_input(frequenties);
    teken_horizontaal(frequenties);
    teken_vertikaal(frequenties);
    return 0;
}

void print_lijn(char c, int lengte){
    int i;
    for(i = 0; i < lengte; i++){
        printf("%c",c);
    }
}

void teken_horizontaal(const int frequenties[]){
    int i;
    for(i = 0; i < 26; i++){
        printf("\n%c:   ",('a'+i));
        print_lijn('*',frequenties[i]);
    }
}
```

```
void teken_vertikaal(const int frequenties[]){
    int grootste_frequentie = 0;
    int i;
    for(i = 0; i < 26; i++){
        if(grootste_frequentie < frequenties[i]){
            grootste_frequentie = frequenties[i];
        }
    }

    for(i = grootste_frequentie; i > 0; i--){
        printf("\n");
        int k;
        for(k = 0; k < 26; k++){
            if(frequenties[k] >= i){
                printf("%c",'a'+k);
            }
            else{
                printf(" ");
            }
        }
    }
}

int is_kleine_letter(char c){
    return ('a' <= c && c <= 'z');
}

int is_grote_letter(char c){
    return ('A' <= c && c <= 'Z');
}

void lees_input(int frequenties[]){
    for(i=0 ; i<26 ;i++){
        frequenties[i]=0;
    }
    char letter;
    scanf("%c",&letter);
    while(letter != '$'){
        if(is_kleine_letter(letter)){
            frequenties[letter-'a']++;
        }
        else if(is_grote_letter(letter)){
            frequenties[letter-'A']++;
        }
        scanf("%c",&letter);
    }
}
```