



RADBOUD UNIVERSITEIT NIJMEGEN
AFDELING STERRENKUNDE
JÖRG R. HÖRANDEL
JAN VAN ROESTEL

Programmeren 1 – 2014/15 Computer Practicum 6 – 15.12.2014

De antwoorden moeten voor 05.01.2015 op blackboard ingediend worden!

Schrijf in elk programma op de eerste regel uw naam en uw studentnummer als commentaar:

```
/* <naam> <studentnummer> */  
#include <stdio.h>
```

Taak 15 (3 punten)

Bereken de Fibonacci nummer f_n . De Fibonacci nummers zijn gedefinieerd door

$$f_0 = 0$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2}$$

Lees n in de hoofdprogramma in en roep de functie fibonacci. Schrijf de recursieve functie fibonacci(n).

Bereken de Fibonacci nummers f_2 , f_{15} en f_{25} .

Taak 16 (3 punten)

Een eendimensionaal veld van tekens (zoals `char text[80]`) noemen wij ook string (character chain). Schrijf een functie *stringsort* die de tekens binnen een string in alfabetische volgorde brengt. A..Z zijn voor a..z. En space is voor A.

```
int stringsort( ..)
```

De return waarde van stringsort zal 0 zijn als er alleen tekens A..Z en a..z binnen de string zijn, en 1 als er nog andere tekens zijn zoals 0..9 of ? ! of space. Voor de berekening van de return waarde is het nuttig om het maximum te berekenen. Gebruik

```
#define max(A,B) ( (A) > (B) ? (A) : (B) )
```

daarvoor. Dit is geen functie. De compiler vervangt `max(A,B)` met `(...)`.

Roep de functie stringsort in een hoofdprogramma met de volgende string: "Dit is een test met HOOFDLETTERS en kleine letters en mijn student nummer is 1234567". U hoofdprogramma print eerst de originele string uit en daarnaar de gesorteerde string. Ook de return waarde van stringsort moet uitgeprint worden.

Taak 17 (5 punten)

De studentenadministratie heeft een lijst met de voornamen, de achternamen en de student nummers van een aantal studenten:

```
#define MAXSTUDENT 6
```

De namen zijn maximaal

```
#define MAXLEN 80
```

tekens lang.

Gebruik de structuur

```
typedef struct{  
    char voornaam[MAXLEN];  
    char achternaam[MAXLEN];  
    int student_nummer;  
} Student;
```

in uw programma. De data zijn in de variabelen

```
int main()  
{  
    Student studenten[MAXSTUDENT];  
    .....  
}
```

opgeslagen.

Schrijf een programma om de achternamen in alfabetisch volgorde te brengen. Schrijf een functie *print_student* om de voornaam, de achternaam en de student nummer van een student af te drukken.

Print in uw programma de originele lijst en de gesorteerde lijst uit.

Om de namen te sorteren moet u beslissen of een naam voor of na een tweede naam in een alfabetisch volgorde staat. Daarvoor gebruiken wij de functie

```
int avoorb(char *a, char *b)
```

Schrijf deze functie: hij geeft 1 terug als de naam in *a voor de naam in *b is, en 0 anders.

In het hoofdprogramma kunt u dan de beslissing

```
if( !avoorb(studenten[i].achternaam, studenten[j].achternaam) )
```

gebruiken.

Gebruik de volgende namen voor een test van uw programma:

Bram Timmermans 1234567

Julia Dijkgraaf 1111111

Sophie Jansen 2222222

Tess Groot 3333333

Levi Nelemans 4444444

uw eigen voornaam uw eigen achternaam, uw eigen student nummer

Een goede methode om de variabelen te initialiseren is

```
strcpy( studenten[0].voornaam, "Bram" );
```

Let op: studenten[0].voornaam = "Bram" werkt NIET in C. Wij kunnen een string NIET zo toewijzen. Wij moeten strcpy gebruiken.

Een gelukkig kerstfeest en een prettig nieuwjaar!