



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

Programming 1 – 2014/15 Computer Practicum 6 – 15.12.2014

The answers should be handed in before 05.01.2015 on blackboard!

Write as a comment on the first line your name and student number:

```
/* <name> <student number> */  
#include <stdio.h>
```

Task 15 (3 points)

Calculate the Fibonacci number f_n . The Fibonacci numbers are defined by

$$f_0 = 0$$

$$f_1 = 1$$

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

Use n as input for the main program and call a function `fibonacci`. Write the recursive function `fibonacci(n)`.

Calculate the Fibonacci numbers f_2 , f_{15} and f_{25} .

Task 16 (3 points)

A one dimensional array of characters (for example `char text[80]`) is also called a string (character chain). Write a function `stringsort` that sorts the characters in such a string in alphabetical order. A..Z before a..z and white space before A.

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

The return value of `stringsort` should be zero when only characters A..Z and a..z are contained in the string, and 1 if other characters are found in the string, like 0..9 or ? ! or white space. For the calculation of the return value, it is useful to calculate the maximum value in the string. Use

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

for this. This is not a function, but the compiler substitutes `max(A,B)` met `((A) > (B) ? (A) : (B))`.

Call this function `stringsort` in the main program using the following string: "Dit is een test met HOOFDLETTERS en kleine letters en mijn student nummer is 1234567". Your main program should first print the original string and then then the sorted string. The return value from `stringsort` should also be printed.

Task 17 (5 points)

The student administration has a list with first names, last names and student numbers:

```
#define MAXSTUDENT 6
```

The names have a maximum length of

```
#define MAXLEN 80
```

characters.

Use a struct

```
typedef struct {  
    char firstname[MAXLEN];  
    char lastname[MAXLEN];  
    int student_number;  
} Student;
```

in your program. The data are save in the variables:

```

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

```

Write a program to sort the last names on alphabetical order. Write a function *print_student* which prints the first name, last name and student number for a student.

Have your main program print the original list and the sorted list.

To sort the names you have to decide if a name occurs before or after a second name in the alphabet. For this we use the function

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

Write this function: it returns 1 if the name in *a occurs before the name in *b, else it returns a 0.

In the main program you can then use:

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

Use the following name for a test of you program:

Bram Timmermans 1234567

Julia Dijkgraaf 1111111

Sophie Jansen 2222222

Tess Groot 3333333

Levi Nelemans 4444444

Your own first name your own last name your own student number

A good method to initialize variables is

```
strcpy( students[0].firstname, "Bram" );
```

Warning: `students[0].firstname = "Bram"` does NOT work in C. You cannot assign a value to a string like this. You have to use `strcpy`.

Merry Christmas and a happy new year!