```
/*****************************************************************************/
/* Title: Uebung 3                                                           */
/* Description: Grundlagen                                                   */
/*                                                                           */
/* Creator:                                                                  */
/* Matr.No: s721011, s782688                                                 */
/* Time of creation:                                                         */
/* Time of modification:                                                     */
/* Compile options: gcc -Wall u3.c -o u3                                     */
/*****************************************************************************/

#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <errno.h>
#include <sys/wait.h>
#include <ctype.h>
#include <sys/time.h>
#define FELDMAX 100

void sortiere(int feld[], int anz);
void zusamenfuegen(int fein[], int faus[], int anz);
void ausgabe(int feld[], int anz);
int fuellen (int feld[],int anz);
long int time_diff(struct timeval begin, struct timeval end);

/*****************************************************************************/
void sortiere(int feld[],int anz)
{       // sortiert feld[] der anz gross ist
        int i,temp;
        while (anz--)
        for(i = 1 ; i<= anz ; i++)
                if(feld[i-1]> feld[i])
                {
                        temp = feld[i];
                        feld[i] = feld[i-1];
                        feld[i-1] = temp;
                }
}

/*****************************************************************************/
void zusamenfuegen(int fein[], int faus[], int anz)
{       // mischen die 1. & 2.Haelfte in fein[], Ergebnis wird in faus[] gespeischert
        int i, j, k;
        i = j = 0;
        for (k = 0;k < anz; k++)
        {

                if (fein[i] < fein[anz/2 + j])
                {
                        faus[k] = fein[i];
                        i++;
                        if (i >= anz/2)
                                i = i +anz/2;
```

```
                }
                else
                {
                        faus[k] = fein[anz/2 + j];
                        j++;
                        if (j >= anz)
                                j = j -(anz/2);

                }
        }
}

/*****************************************************************/
void ausgabe(int feld[], int anz)
{       // gibt die Liste der Elemente vom Array feld auf Bildschrim
        int i = 0;
        /*      for(i =0 ; i< anz ; i++){
                        printf(" %4d" , feld[i] );
                        if(i % 4 == 0 && i != 0)
                        printf("\n");
                }
                printf("\n");
        */

        for (i=0; i<anz/4 ;i++)
        {
                printf(" %4d. Zahl: %5d l %4d. Zahl: %5d l %4d. Zahl: %5d l %4d. Zahl: %5d \n",i+1,
feld[i], i+anz/4+1 , feld[i+anz/4] , i+anz/2+1 , feld[i+anz/2] , i+anz*3/4+1 , feld[i+anz*3/4]);
        }
}

/*********************************************************************/
int fuellen (int feld[],int anz)
{
        int i;
        for (i = 0; i < anz; i++)
        {
                feld[i] = rand()%10000;
        }
        return 0;
}

/*************************************************************************************/
long int time_diff(struct timeval begin, struct timeval end)
{       //gibt die zeit diffenz zurck im IntegerFormat
        long int a = ((end.tv_sec * 1000000 + end.tv_usec) - (begin.tv_sec * 1000000 +
begin.tv_usec));
        return a;
}

/************************** MAIN *********************************/
int main()
{
        int feld[FELDMAX];
        int feldaus[FELDMAX];
        int pid;
```

```
int fd[2];
// int i;
//fd[0] is for reading
////fd[1] is for writtig

struct timeval t1,t2,t3,t4;
srand( t1.tv_usec * t1.tv_sec);
gettimeofday(&t1,NULL);
fuellen(feld + FELDMAX/2, FELDMAX/2);
gettimeofday(&t2,NULL);
long int a = time_diff(t1,t2);
printf("Zeiten zum fuellen 2Â° haelfte der felder :%ld \n",a);
fuellen(feld, FELDMAX/2);
gettimeofday(&t4,NULL);
long int b = time_diff(t1, t4);
printf("Zeiten zum fuellen 1Â° haelfte der felder :%ld \n ",b);

/* pipe erzeugen */
if (pipe(fd) < 0)
{
        perror("Fehler beim Einrichten der Pipe!\n");
        exit(1);
}
switch(pid =fork())
{
        case -1:        perror("Fehler beim Fork-Aufruf! \n");
                        exit(1);
                        break;

        case 0:         // Kindprozess

                        sortiere(feld + FELDMAX/2,FELDMAX/2);
                        // close(fd[0]); return 0 wenn succes --> -1 wenn error
                        if(close(fd[0])<0)
                        {
                                perror("close fd[0]");
                                return (1);
                        }
                        if(write(fd[1],feld+FELDMAX/2,FELDMAX/2*sizeof(int)) < 0 )
                        {
                                perror("write fd[1]");
                                exit(0);
                        }
                        if (close(fd[1]) < 0)
                        {
                                perror("close_fd1");
                                return(1);
                        }
                        break;

        default: // Eltern-Prozess
                        sortiere(feld,FELDMAX/2);
                        // close(fd[1]);
                        if(close(fd[1])<0)
                        {
                                perror("close_fd[1]");
```

```
                return (1);
            }
            if(read(fd[0],feld+FELDMAX/2,FELDMAX/2*sizeof(int)) < 0)
            {
                perror("read_fd[0]");
                return (1);
            }
            if(close(fd[0])<0)
            {
                perror("close_fd[1]");
                return (1);
            }
            wait(0);
            zusamenfuegen(feld,feldaus,FELDMAX);

            printf("\n********** Teil 1 **********\n");
            ausgabe(feld,FELDMAX/2);

            printf("\n********** Teil 2 **********\n");
            ausgabe(feld+FELDMAX/2, FELDMAX/2);

            printf("\n*########## Verschmelzung ##########*\n");
            ausgabe(feldaus,FELDMAX);
            break;
        }
        return 0;
}
```