```
/*****************************************************************************/
/* Title: Uebung 2                                                           */
/* Description: Grundlagen                                                   */
/*                                                                           */
/* Creator:                                                                  */
/* Matr.No: s721011, s782688                                                 */
/* Time of creation:                                                         */
/* Time of modification:                                                     */
/* Compile options: gcc -Wall u2.c -o u2                                     */
/*****************************************************************************/
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <errno.h>
#include <sys/wait.h>

#define ZEILENLAENGE 20
#define MAXZEILEN 9900
#define MAX_ANZAHL 10

char daten[MAXZEILEN][ZEILENLAENGE];
char keys[MAXZEILEN][MAX_ANZAHL + 1];
int laenge;
/*******************************************************************************************/
void loeschen(int idx)
{
        int i;
        if (idx == laenge -1)                   //der Array beginnt mit '0'
                laenge--;
        else            //loesche die doppelten namen und lasse nur eine in die datei
        {
                for (i = idx; i < laenge-1; i++)
                {
                        strcpy(daten[i], daten[i+1]);
                        strcpy(keys[i], keys[i+1]);
                }
                laenge--;
        }
}
/*****************************************************************************/
int sortieren()
{
        int i,j, cmp;
        char temp[ZEILENLAENGE];        //Array
        for(i = 0; i < laenge -1; i++)
        {
                for(j = i + 1; j < laenge; j++)
                {
                        cmp = strcmp(daten[i],daten[j]);//sortiert die name in die datei nlakey01.txt
                        if( cmp > 0 )
                        {
                                strcpy(temp, daten[i]);
                                strcpy(daten[i], daten[j]);
                                strcpy(daten[j], temp);
```

```
                                strcpy(temp, keys[i]);
                                strcpy(keys[i], keys[j]);
                                strcpy(keys[j], temp);
                        }
                        else if (cmp == 0)       //gleiche Namen loeschen
                        {
                                loeschen(j);
                                j--;
                        }
                }
        }
return 0;
}
/*****************************************************************************/
void random_string (char * string, unsigned Laenge)
{
        /* ASCII-Zeichen 33-126 */
        int i;
        for (i = 0; i <Laenge; ++ i)
        {
                string [i] = rand ()% 94 + 33;  //Ezeugt eine string der enthaelt zufall ASCII zeichen
        }
string [i] = '\0';
}
/*****************************************************************************/
void createKeys()
{
        int i = 0;
        srand (time(NULL));
        do
        {
                random_string (keys[i], 10);   //erzeugt fur jeden namen ein keys random
                i++;
        }
        while(i < laenge);
}
/*****************************************************************************/
int readFile(char * name)
{       //lesen zeileweise die daten einer Datei und spiechern in der variable daten [][]
        FILE *fp;
        if ((fp = fopen(name ,"r")) == NULL)
        {
                fprintf (stderr,"Datei %s konnte nicht  geoeffnet werden ",name );
                return -1;
        }
        else
        {
                laenge = 0;
                while(fgets(daten[laenge], ZEILENLAENGE, fp))
                {
                        daten[laenge][strlen(daten[laenge])-1] = '\0';
                        laenge++;
                }
        }
fclose(fp);
return 0;
```

```
}
//*****************************************************************************************/
int writeFile(char * name)
{       //schreiben die sortierte Namen und seine Keys in einer Ausgabesdatei
        FILE *fp;
        int i;

        if ((fp = fopen(name ,"w")) == NULL)
        {
                fprintf (stderr,"Datei %s konnte nicht geoeffnet werden ",name );
                return -1;
        }
        else
        {
                i = 0;
                do
                {
                        fputs(daten[i],fp);
                        fputs(":",fp);
                        fputs(keys[i],fp);
                        fputs("\n",fp);
                        i++;
                }
                while(i < laenge);
        }
fclose(fp);
return 0;
}
/*****************************************************************/
int main(void)
{
        int pid;

        switch (pid = fork ())
        {
                case -1:        perror("FehLer beim der fork \n");
                                return -1;
                                break;

                case 0:         printf("kind prozess mit pid : %d \n\n" ,getpid());
                                readFile("namensliste.txt");
                                createKeys();                   //mit random
                                sortieren();                    //sortiert und loescht gleiche Namen
                                writeFile("nlakey01.txt");      //schreibe in neue datei name+:+key
                                break;

                default:        printf("Eltern prozess mit pid : %d \n\n" ,getppid());
                                break;
        }
return 0;
}
```