

Lösung Praktikum 5

1 Aufgabe 1: Das Josephus-Spiel (Ringliste)

```
#include <stdio.h>

//Datenstruktur für die Listenelemente
//nummer gibt die Position im Kreis
//name ist der Name der Person
//*naechster zeigt auf das nächste Listenelement
//Achtung Ringliste: letztes Element zeigt auf erstes
struct person_t
{
    int nummer;
    char name[20];
    struct person_t *naechster;
};

//Funktion zum Anlegen der Liste
//d gibt die Anzahl der Personen im Kreis
//*liste ist Zeiger auf den Anfang der Liste
struct person_t *anlegen(int d, struct person_t *liste)
{
    //ein Zeiger auf Listenelement, um das aktuelle Element
    //zu merken
    //ein Zeiger auf Listenelement, um neue Elemente anzulegen
    struct person_t *aktuell = NULL;
    struct person_t *neu = NULL;
    //Indexzähler
    int i;

    //Speicher für das erste Listenelement reservieren
    liste = (struct person_t *)malloc(sizeof(struct person_t));

    //Listenelement belegen:
    //1. Person im Kreis
    liste->nummer = 1;
    //Namen der Person einlesen
    printf("Gib den Namen der 1. Person ein: ");
    scanf("%s", &(liste->name));
    //Letztes Element zeigt auf erstes, in diesem Fall
    //also auf sich selbst
    liste->naechster = liste;
```

```

//das aktuelle Listenelement ist jetzt das erste
aktuell = liste;

//restliche Listenelemente erzeugen
for (i = 2; i <= d; i++)
{
    //Speicher reservieren
    neu = (struct person_t *)malloc(sizeof(struct person_t));

    //Nummer im Kreis angeben
    neu->nummer = i;
    //Name der Person einlesen
    printf("Gib den Namen der %d. Person ein: ", i);
    scanf("%s", &(neu->name));
    //bei neuem Element ist das nächste Element stets das erste
    neu->naechster = liste;

    //der Nachfolger des aktuellen Elements wird das neue
    aktuell->naechster = neu;
    //jetzt kann das gerade eingefuegte Element aktuelles
    //Element werden
    aktuell = neu;
}

//Liste zurueckgeben
return liste;
}

//Funktion zum Ausgeben derListe
//*liste Zeiger auf Liste
void ausgeben(struct person_t *liste)
{
    //lokale Variable fuer aktuelles Element
    //am Anfang auf Anfang der Liste
    struct person_t *aktuell = liste;

    //lese bis zum Ende
    //wenn der Nachfolger des aktuellen Elements der Listenanfang
    //ist, sind alle Elemente gelesen worden
    do
    {
        //gib Nummer und Name der Person aus
        printf("Nr. %d %s\n", aktuell->nummer, aktuell->name);
        //setze aktuelles Element auf das naechste
        aktuell = aktuell->naechster;
    }while (aktuell != liste);
}

//Funktion zum Auszaehlen
//n gibt an der jeweils wievielte ausgezaehlt werden soll

```

```

// *liste ist Zeiger auf Listenanfang
void auszaehlen(int n, struct person_t *liste)
{
    // lokale Variable fuer aktuelles Element, am Anfang Kopf der Liste
    struct person_t *aktuell = liste;
    struct person_t *temp;

    // Indexzaehler
    int i;

    // zaehle aus bis nur noch einer uebrig ist
    // in diesem Fall zeigt der Zeiger auf das naechste Element
    // auf das Element selbst
    do
    {
        // n Schritte weitergehen
        for (i = 1; i < n; i++)
            aktuell = aktuell->naechster;
        // Angeben wer aussortiert wird
        printf("Nr. %d %s wird aussortiert!\n",
            aktuell->naechster->nummer, aktuell->naechster->name);
        // Aussortiertes Element ausketten, d.h. Zeiger auf das
        // uebernaechste Element setzen
        // temp merkt aussortiertes Element
        temp = aktuell->naechster;
        aktuell->naechster = aktuell->naechster->naechster;
        // Speicher des aussortierten Elements freigeben
        free(temp);
    } while(aktuell != aktuell->naechster);
}

int main()
{
    // Variablen fuer Anzahl der Personen, das Auszaehlen und die Liste
    int anzahl;
    int aussortieren;
    struct person_t *liste = NULL;

    // Anzahl der Personen einlesen
    printf("Wieviele Personen stehen im Kreis? ");
    scanf("%d", &anzahl);

    // Auszaehlweise einlesen
    printf("Die wievielte Person soll ausgezaehlt werden? ");
    scanf("%d", &aussortieren);

    // Liste anlegen und ausgeben
    liste = anlegen(anzahl, liste);
    ausgeben(liste);

    // Aussortieren

```

```
printf("Jetzt wird aussortiert:\n");  
auszaehlen(aussortieren, liste);  
}
```