Entwicklungsdokumentation

Dateien:

```
typedef struct recept {
    char* name;
    char* ossze;

    char* elkeszit;
    int osszcucc;
    struct recept* next;
}Recept;
```

Dieses Struct ist die Base der verketteten Liste.

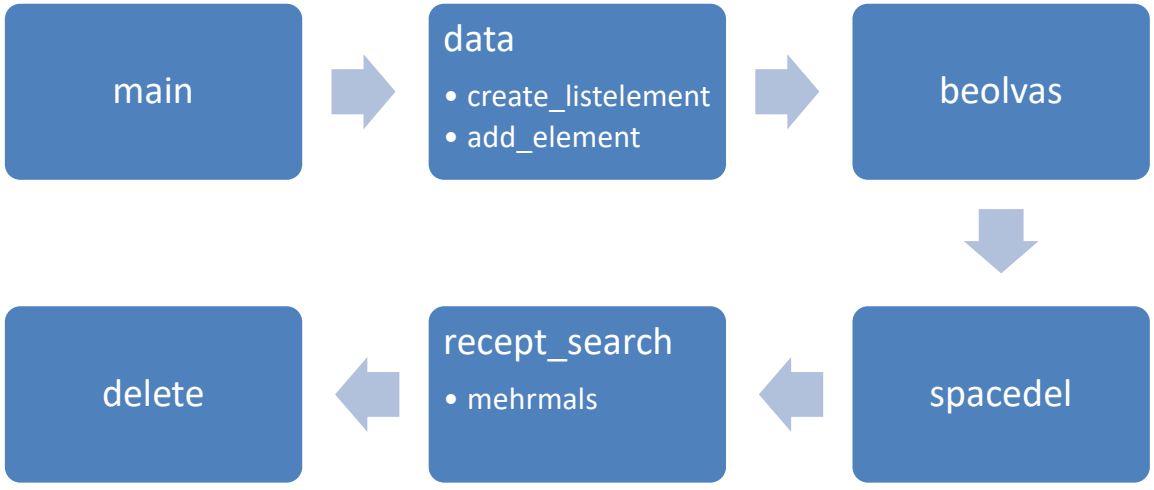
```
typedef struct recept {
    char* name;
    char* ossze;
    char* elkeszit;
    int osszcucc;
    struct recept* next;
}Recept;
```

```
Fájl  Szerkesztés  Formátum  Nézet  Súgó
rántotta;
tojás,só;
keverd össze a tojást majd süssd ki;
2;
limonádé;
víz,cukor,citrom;
```

Es behandelt die Dateien aus einem File.

In den File sind die Dateien in der Reihenfolge wie es auf dem Bild zu sehen ist. Sie werden immer mit einem „ ; “ getrennt.

Der Aufbau des Programmes ist auf dem Diagramm zu sehen:




```
int main()
```

Es sorgt für die Erzeugung alle anderen Programme. Es ist wichtig dass hier werden die

Ungarische Charakteren (nur auf Windows) eingestellt mit dem
SetConsoleCP(1250);
SetConsoleOutputCP(1250);
es zu verwirklichen muss man die Windows.h und die locale.h includieren.
Wenn keine Rezepte gut sind liefert es „bad luck“ zurück.

```
Recept* data(Recept* head);
```

Diese Funktion liest die Rezepte aus dem gegebenen File ein und erzeugt eine verkettete Liste von ihnen. Wenn der File nicht geöffnet werden kann, schreibt das Programm ein Fehler auf stderr. Es liest alles von Charakter zu Charakter ein mit dem Funktion fgetc() Vorsicht es liefert ein int typ Variable zurück! Den ganzen Struct (siehe später) wird in einen großen String eingelesen und dann es wurde verteilt in 4 teilen und in einen Struct kopiert der mit

```
⊞Recept* create_listelement()
```

erzeugt wird. Der sorgt für die dynamische allokieren des Structes.
Dieses aufgefühltes Struct wird mit

```
void add_element(Recept** head, Recept* e) { ... }
```

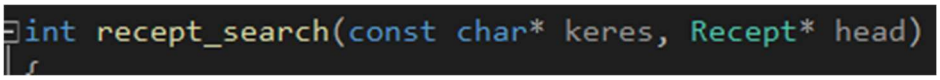
In eine verkettete Liste hingefügt. Es stellt die Pointer der Letzte Elemente auf die Pointer des zurzeit bearbeiteten Structes ein.
Wenn die Datenstruktur schon fertig ist wird das Programm auf das Userinput warten

```
char* beolvas(char*string)
```

Diese Funktion sorgt für die Einlesen der Zutaten. Es wird mit einem String initialisiert mit einen ‚\0‘ und immer realloziert bis ein „\n“(enter) kommt. Vorsicht realloc kann ein NULL zurückliefern und deshalb es muss behandelt werden auf stderr schreibt es diese Probleme aus.

```
char* str_spacedel(char* str)
```

Sie entfernt alle ‚Space‘ Charakteren mit einem einfachen while zyklus um die komparieren einfacher zu machen.



Diese Funktion bekommt die Eingabe der User und die ersten Elemente der verketteten Liste.

Sie teilt die Zutaten der einzigen Structen auf (immer wo, steht). Sie sucht, ob dieses Teil in den Input zu finden ist wenn ja erhöht sie die Summe der gefundene Zutaten, wenn es durch den String gegangen hat. Vergleichung mit dem Wert der Structes (anzahl_der_zutaten). Wenn es einen MATCH hat, liefert es 1 zurück. Es wurde in main genau so viel mal gerufen wie viele Rezepte vorhanden sind. IN main werden die Rezepte ausgeschrieben.