

Recepteskonyv

Generated by Doxygen 1.12.0

1 Data Structure Index	1
1.1 Data Structures	1
2 File Index	3
2.1 File List	3
3 Data Structure Documentation	5
3.1 Ingridient Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Field Documentation	5
3.1.2.1 IngName	5
3.1.2.2 MType	5
3.1.2.3 NextNode	6
3.2 Recipe Struct Reference	6
3.2.1 Detailed Description	6
3.2.2 Field Documentation	6
3.2.2.1 blsFavorite	6
3.2.2.2 NextNode	6
3.2.2.3 RecIng	6
3.2.2.4 RecName	7
3.3 RecipeIngridient Struct Reference	7
3.3.1 Detailed Description	7
3.3.2 Field Documentation	7
3.3.2.1 IngAmount	7
3.3.2.2 IngData	7
3.3.2.3 NextNode	7
4 File Documentation	9
4.1 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Ingridient.c File Reference	9
4.1.1 Function Documentation	9
4.1.1.1 AddIngridientTo()	9
4.1.1.2 AddOrCreatelng()	10
4.1.1.3 CreateIngridient()	10
4.1.1.4 DeleteAllIng()	10
4.1.1.5 DeleteIngridient()	10
4.1.1.6 FindIngridientByName()	11
4.1.1.7 PrintAllIng()	11
4.1.1.8 PrintIngNameByIndex()	11
4.2 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Ingridient.h File Reference	12
4.2.1 Typedef Documentation	12
4.2.1.1 Ingridient	12
4.2.1.2 MeasurementType	12
4.2.1.3 Recipe	12

4.2.2 Enumeration Type Documentation	12
4.2.2.1 MeasurementType	12
4.2.3 Function Documentation	13
4.2.3.1 AddIngridientTo()	13
4.2.3.2 AddOrCreateIng()	13
4.2.3.3 CreateIngridient()	13
4.2.3.4 DeleteAllIng()	14
4.2.3.5 DeleteIngridient()	14
4.2.3.6 FindIngridientByName()	14
4.2.3.7 PrintAllIng()	14
4.2.3.8 PrintIngNameByIndex()	15
4.3 Ingridient.h	15
4.4 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/input.c File Reference	16
4.4.1 Function Documentation	16
4.4.1.1 IngInput()	16
4.4.1.2 IngTypeDenominator()	16
4.4.1.3 IngTypeEnumrator()	16
4.4.1.4 IsFavChecker()	17
4.4.1.5 RecInput()	17
4.4.1.6 StrKiller()	17
4.5 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/input.h File Reference	17
4.5.1 Typedef Documentation	18
4.5.1.1 IngInputState	18
4.5.1.2 RecInputState	18
4.5.2 Enumeration Type Documentation	18
4.5.2.1 IngInputState	18
4.5.2.2 RecInputState	18
4.5.3 Function Documentation	19
4.5.3.1 IngInput()	19
4.5.3.2 IngTypeDenominator()	19
4.5.3.3 IngTypeEnumrator()	19
4.5.3.4 IsFavChecker()	19
4.5.3.5 RecInput()	20
4.5.3.6 StrKiller()	20
4.6 input.h	20
4.7 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/main.c File Reference	21
4.7.1 Function Documentation	21
4.7.1.1 main()	21
4.8 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Menu.c File Reference	21
4.8.1 Function Documentation	21
4.8.1.1 FavMenu()	21
4.8.1.2 IngManageMenu()	22

4.8.1.3 LogoPrinter()	22
4.8.1.4 MainMenu()	22
4.8.1.5 RandomRecipe()	22
4.8.1.6 RecipeMenu()	23
4.8.1.7 SearchMenu()	23
4.9 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Menu.h File Reference	23
4.9.1 Function Documentation	24
4.9.1.1 FavMenu()	24
4.9.1.2 IngManageMenu()	24
4.9.1.3 LogoPrinter()	24
4.9.1.4 MainMenu()	24
4.9.1.5 RandomRecipe()	24
4.9.1.6 RecipeMenu()	25
4.9.1.7 SearchMenu()	25
4.10 Menu.h	25
4.11 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Recipe.c File Reference	26
4.11.1 Function Documentation	26
4.11.1.1 AddOrCreateRecipe()	26
4.11.1.2 AddRecipeFromIO()	26
4.11.1.3 AddRecipeFromIOsFav()	27
4.11.1.4 AddRecipeTo()	27
4.11.1.5 AssignIngridients()	27
4.11.1.6 BoolFavConverter()	27
4.11.1.7 CreateRecipe()	28
4.11.1.8 DeleteAllRecipe()	28
4.11.1.9 DeleteRecipe()	28
4.11.1.10 FindRecipeByName()	28
4.11.1.11 PrintAllRecipes()	29
4.11.1.12 PrintFavRecipes()	29
4.11.1.13 PrintRecipeByIng()	29
4.11.1.14 RecipeListLen()	29
4.11.1.15 RecipePrint()	30
4.11.1.16 SetFavorite()	30
4.12 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Recipe.h File Reference	30
4.12.1 Typedef Documentation	31
4.12.1.1 Recipe	31
4.12.2 Function Documentation	31
4.12.2.1 AddOrCreateRecipe()	31
4.12.2.2 AddRecipeFromIO()	31
4.12.2.3 AddRecipeFromIOsFav()	32
4.12.2.4 AddRecipeTo()	32
4.12.2.5 AssignIngridients()	32

4.12.2.6 BoolFavConverter()	32
4.12.2.7 CreateRecipe()	33
4.12.2.8 DeleteAllRecipe()	33
4.12.2.9 DeleteRecipe()	33
4.12.2.10 FindRecipeByName()	33
4.12.2.11 PrintAllRecipes()	34
4.12.2.12 PrintFavRecipes()	34
4.12.2.13 PrintRecipeByIng()	34
4.12.2.14 RecipeListLen()	34
4.12.2.15 RecipePrint()	35
4.12.2.16 SetFavorite()	35
4.13 Recipe.h	35
4.14 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/RecipeIngridient.c File Reference	36
4.14.1 Function Documentation	36
4.14.1.1 AddRITo()	36
4.14.1.2 CreateRecipeIngridient()	37
4.14.1.3 DeleteAllRecIng()	37
4.14.1.4 DeleteRI()	37
4.14.1.5 FindRecipeIngridientByName()	37
4.15 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/RecipeIngridient.h File Reference	38
4.15.1 Typedef Documentation	38
4.15.1.1 RecipeIngridient	38
4.15.2 Function Documentation	38
4.15.2.1 AddRITo()	38
4.15.2.2 CreateRecipeIngridient()	39
4.15.2.3 DeleteAllRecIng()	39
4.15.2.4 DeleteRI()	39
4.15.2.5 FindRecipeIngridientByName()	39
4.16 RecipeIngridient.h	40
Index	41

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Ingridient	5
Recipe	6
RecipeIngridient	7

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ Ingridient.c	9
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ Ingridient.h	12
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ input.c	16
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ input.h	17
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ main.c	21
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ Menu.c	21
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ Menu.h	23
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ Recipe.c	26
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ Recipe.h	30
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ RecipeIngridient.c	36
C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/ RecipeIngridient.h	38

Chapter 3

Data Structure Documentation

3.1 Ingridient Struct Reference

```
#include <Ingridient.h>
```

Data Fields

- char [IngName](#) [40]
- [MeasurementType](#) [MType](#)
- struct [Ingridient](#) * [NextNode](#)

3.1.1 Detailed Description

Ingridient struct that stores and [Ingridient](#) name, Mesurment Type of the ingridient in the form of an enum and an ingridient pointer that is used to make a linked list.

3.1.2 Field Documentation

3.1.2.1 IngName

```
char Ingridient::IngName[40]
```

The ingridients name.

3.1.2.2 MType

```
MeasurementType Ingridient::MType
```

The type of the ingridnient.

3.1.2.3 NextNode

```
struct Ingridient* Ingridient::NextNode
```

The pointer to the next [Ingridient](#).

The documentation for this struct was generated from the following file:

- C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/[Ingridient.h](#)

3.2 Recipe Struct Reference

```
#include <Recipe.h>
```

Data Fields

- char [RecName](#) [100]
- struct [Recipe](#) * [NextNode](#)
- [RecipeIngridient](#) * [RecIng](#)
- bool [blsFavorite](#)

3.2.1 Detailed Description

The struct that stores recipe values.

3.2.2 Field Documentation

3.2.2.1 blsFavorite

```
bool Recipe::blsFavorite
```

A boolean value that dictates if the recipe is Favorite or not.

3.2.2.2 NextNode

```
struct Recipe* Recipe::NextNode
```

The pointer to the next recipe in the recipe linked list.

3.2.2.3 RecIng

```
RecipeIngridient* Recipe::RecIng
```

The pointer to the first element of the RecipeIngridient linked list.

3.2.2.4 RecName

```
char Recipe::RecName[100]
```

The name of the recipe

The documentation for this struct was generated from the following file:

- C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/[Recipe.h](#)

3.3 RecipeIngridient Struct Reference

```
#include <RecipeIngridient.h>
```

Data Fields

- int [IngAmount](#)
- [Ingridient](#) * [IngData](#)
- struct [RecipeIngridient](#) * [NextNode](#)

3.3.1 Detailed Description

A wrapper struct that connects the [Ingridient](#) struct and the [Recipe](#) struct and it stores ingridient amount. The [IngData](#) points to the selected [Ingridient](#). The [NextNode](#) is used to point to the next element of the [RecipeIngridient](#) linked list.

3.3.2 Field Documentation

3.3.2.1 IngAmount

```
int RecipeIngridient::IngAmount
```

The amount of the selected ingridient.

3.3.2.2 IngData

```
Ingridient* RecipeIngridient::IngData
```

The pointer of the selected ingridient.

3.3.2.3 NextNode

```
struct RecipeIngridient* RecipeIngridient::NextNode
```

The pointer to then next [RecipeIngridient](#) in the chained list.

The documentation for this struct was generated from the following file:

- C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/[RecipeIngridient.h](#)

Chapter 4

File Documentation

4.1 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Ingridient.c File Reference

```
#include "Ingridient.h"
#include <stdio.h>
#include "Recipe.h"
#include "debugmalloc.h"
```

Functions

- [Ingridient](#) * [CreateIngridient](#) (char *Name, [MeasurementType](#) Type)
- void [AddIngridientTo](#) ([Ingridient](#) *list, [Ingridient](#) *ToAdd)
- void [AddOrCreateIng](#) ([Ingridient](#) **list, [Ingridient](#) *ToAdd)
- void [DeleteIngridient](#) ([Ingridient](#) **listPtr, [Ingridient](#) *ToDelete, [Recipe](#) **RecListPtr)
- [Ingridient](#) * [FindIngridientByName](#) ([Ingridient](#) *list, char *IngName)
- void [PrintIngNameByIndex](#) ([Ingridient](#) *list, int IngridientIndex)
- void [PrintAllIng](#) ([Ingridient](#) *list)
- void [DeleteAllIng](#) ([Ingridient](#) **listPtr, [Recipe](#) **RecListPtr)

4.1.1 Function Documentation

4.1.1.1 AddIngridientTo()

```
void AddIngridientTo (
    Ingridient * list,
    Ingridient * ToAdd)
```

Adds an [Ingridient](#) to an already existing [Ingridient](#) linked list.

Parameters

<i>list</i>	The first address of the list.
<i>ToAdd</i>	The Pointer of Ingridient that will be added to the list.

4.1.1.2 AddOrCreateIng()

```
void AddOrCreateIng (
    Ingridient ** list,
    Ingridient * ToAdd)
```

This function is not used.

Parameters

<i>list</i>	
<i>ToAdd</i>	

4.1.1.3 CreateIngridient()

```
Ingridient * CreateIngridient (
    char * Name,
    MeasurementType Type)
```

Creates an [Ingridient](#) pointer.

Parameters

<i>Name</i>	Name of the ingridient.
<i>Type</i>	Mesurment type of the ingridient.

Returns

[Ingridient](#) pointer.

4.1.1.4 DeleteAllIng()

```
void DeleteAllIng (
    Ingridient ** listPtr,
    Recipe ** RecListPtr)
```

Deletes all ingrideients. This will delete all the recipes too

Parameters

<i>listPtr</i>	The address of the first element of the Recipe linked list.
<i>RecListPtr</i>	The address of the first element of the Ingridient linked list.

4.1.1.5 DeleteIngridient()

```
void DeleteIngridient (
    Ingridient ** listPtr,
    Ingridient * ToDelete,
    Recipe ** RecListPtr)
```

Deletes and element of the linked list.

Parameters

<i>listPtr</i>	Pointer of the first address.
<i>ToDelete</i>	The list element that will be deleted.
<i>RecListPtr</i>	Pointer of the first element of the Recipe linked list.

4.1.1.6 FindIngridientByName()

```
Ingridient * FindIngridientByName (
    Ingridient * list,
    char * IngName)
```

Finds an ingridient by name

Parameters

<i>list</i>	The address of the first element of the ingridient linked list.
<i>IngName</i>	A name of the ingridient that is being searched.

Returns

The address of the [Ingridient](#).

4.1.1.7 PrintAllIng()

```
void PrintAllIng (
    Ingridient * list)
```

Prints all Ingridients.

Parameters

<i>list</i>	The address of the first element of the Ingridient linked list.
-------------	---

4.1.1.8 PrintIngNameByIndex()

```
void PrintIngNameByIndex (
    Ingridient * list,
    int IngridientIndex)
```

Prints the elemnt definded by the index.

Parameters

<i>list</i>	The address first element of the Ingridient linked list.
<i>IngridientIndex</i>	an index.

4.2 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Ingridient.h File Reference

```
#include "debugmalloc.h"
```

Data Structures

- struct [Ingridient](#)

Typedefs

- typedef struct Recipe [Recipe](#)
- typedef enum [MeasurementType](#) [MeasurementType](#)
- typedef struct Ingridient [Ingridient](#)

Enumerations

- enum [MeasurementType](#) { [kilogramm](#) , [liter](#) , [darab](#) }

Functions

- [Ingridient](#) * [CreateIngridient](#) (char *Name, [MeasurementType](#) Type)
- void [AddIngridientTo](#) ([Ingridient](#) *list, [Ingridient](#) *ToAdd)
- void [AddOrCreatelng](#) ([Ingridient](#) **list, [Ingridient](#) *ToAdd)
- void [DeleteIngridient](#) ([Ingridient](#) **listPtr, [Ingridient](#) *ToDelete, [Recipe](#) **RecListPtr)
- [Ingridient](#) * [FindIngridientByName](#) ([Ingridient](#) *list, char *IngName)
- void [PrintIngNameByIndex](#) ([Ingridient](#) *list, int IngridientIndex)
- void [PrintAllIng](#) ([Ingridient](#) *list)
- void [DeleteAllIng](#) ([Ingridient](#) **listPtr, [Recipe](#) **RecListPtr)

4.2.1 Typedef Documentation

4.2.1.1 Ingridient

```
typedef struct Ingridient Ingridient
```

Ingridient struct that stores and [Ingridient](#) name, Mesurment Type of the ingridient in the form of an enum and an ingridient pointer that is used to make a linked list.

4.2.1.2 MeasurementType

```
typedef enum MeasurementType MeasurementType
```

Enum that indicates the ingridient mesurement type.

4.2.1.3 Recipe

```
typedef struct Recipe Recipe
```

4.2.2 Enumeration Type Documentation

4.2.2.1 MeasurementType

```
enum MeasurementType
```

Enum that indicates the ingridient mesurement type.

Enumerator

kilogramm	
liter	
darab	

4.2.3 Function Documentation

4.2.3.1 AddIngridientTo()

```
void AddIngridientTo (
    Ingridient * list,
    Ingridient * ToAdd)
```

Adds an [Ingridient](#) to an already existing [Ingridient](#) linked list.

Parameters

<i>list</i>	The first address of the list.
<i>ToAdd</i>	The Pointer of Ingridient that will be added to the list.

4.2.3.2 AddOrCreateIng()

```
void AddOrCreateIng (
    Ingridient ** list,
    Ingridient * ToAdd)
```

This function is not used.

Parameters

<i>list</i>	
<i>ToAdd</i>	

4.2.3.3 CreateIngridient()

```
Ingridient * CreateIngridient (
    char * Name,
    MeasurementType Type)
```

Creates an [Ingridient](#) pointer.

Parameters

<i>Name</i>	Name of the ingridient.
<i>Type</i>	Mesurment type of the ingridient.

Returns

[Ingridient](#) pointer.

4.2.3.4 DeleteAllIng()

```
void DeleteAllIng (
    Ingridient ** listPtr,
    Recipe ** RecListPtr)
```

Deletes all ingrdeients. This will delete all the recipes too

Parameters

<i>listPtr</i>	The address of the first element of the Recipe linked list.
<i>RecListPtr</i>	The address of the first element of the Ingridient linked list.

4.2.3.5 DeleteIngridient()

```
void DeleteIngridient (
    Ingridient ** listPtr,
    Ingridient * ToDelete,
    Recipe ** RecListPtr)
```

Deletes and element of the linked list.

Parameters

<i>listPtr</i>	Pointer of the first address.
<i>ToDelete</i>	The list element that will be deleted.
<i>RecListPtr</i>	Pointer of the first element of the Recipe linked list.

4.2.3.6 FindIngridientByName()

```
Ingridient * FindIngridientByName (
    Ingridient * list,
    char * IngName)
```

Finds an ingridient by name

Parameters

<i>list</i>	The address of the first element of the ingridient linked list.
<i>IngName</i>	A name of the ingridient that is being searched.

Returns

The address of the [Ingridient](#).

4.2.3.7 PrintAllIng()

```
void PrintAllIng (
    Ingridient * list)
```

Prints all Ingridients.

Parameters

<i>list</i>	The address of the first element of the Ingridient linked list.
-------------	---

4.2.3.8 PrintIngNameByIndex()

```
void PrintIngNameByIndex (
    Ingridient * list,
    int IngridientIndex)
```

Prints the elemnt definded by the index.

Parameters

<i>list</i>	The address first element of the Ingridient linked list.
<i>IngridientIndex</i>	an index.

4.3 Ingridient.h

[Go to the documentation of this file.](#)

```
00001 //
00002 // Created by hejag on 23/11/2024.
00003 //
00004
00005 #ifndef INGRIDIENT_H
00006 #define INGRIDIENT_H
00007 #include "debugmalloc.h"
00008
00009
00010 typedef struct Recipe Recipe;
00011
00015 typedef enum MeasurementType {
00016     kilogramm,
00017     liter,
00018     darab
00019 } MeasurementType;
00020
00024 typedef struct Ingridient {
00028     char IngName[40];
00032     MeasurementType MType;
00036     struct Ingridient *NextNode;
00037 } Ingridient;
00038
00045 Ingridient *CreateIngridient(char *Name, MeasurementType Type);
00046
00052 void AddIngridientTo(Ingridient *list, Ingridient *ToAdd);
00053
00059 void AddOrCreateIng(Ingridient **list, Ingridient *ToAdd);
00060
00067 void DeleteIngridient(Ingridient **listPtr, Ingridient *ToDelete, Recipe **RecListPtr);
00068
00075 Ingridient *FindIngridientByName(Ingridient *list, char *IngName);
00076
00082 void PrintIngNameByIndex(Ingridient *list, int IngridientIndex);
00083
00088 void PrintAllIng(Ingridient *list);
00089
00095 void DeleteAllIng(Ingridient **listPtr, Recipe **RecListPtr);
00096 #endif //INGRIDIENT_H
```

4.4 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/input.c File Reference

```
#include "debugmalloc.h"
#include "input.h"
#include <stdio.h>
#include "Ingridient.h"
#include "Recipe.h"
```

Functions

- [Ingridient](#) * [IngInput](#) ()
- [MeasurementType](#) [IngTypeEnumerator](#) (char *TypeString)
- char * [IngTypeDenumerator](#) ([MeasurementType](#) Mtype)
- void [StrKiller](#) (char *str)
- [Recipe](#) * [RecInput](#) ([Ingridient](#) *list)
- bool [IsFavChecker](#) (char *TypeString)

4.4.1 Function Documentation

4.4.1.1 IngInput()

```
Ingridient * IngInput ()
```

4.4.1.2 IngTypeEnumerator()

```
char * IngTypeEnumerator (
    MeasurementType Mtype)
```

Converts MeasurementType enum to a string.

Parameters

<i>Mtype</i>	MeasurementType enum that will be converted.
--------------	--

Returns

Address of the string.

4.4.1.3 IngTypeEnumerator()

```
MeasurementType IngTypeEnumerator (
    char * TypeString)
```

Converts a string to a MeasurementType enum.

Parameters

<i>TypeString</i>	The that will be converted
-------------------	----------------------------

Returns

MeasurementType.

4.4.1.4 IsFavChecker()

```
bool IsFavChecker (  
    char * TypeString)
```

Converts a string to a true or false value.

Parameters

<i>TypeString</i>	The string that will be converted.
-------------------	------------------------------------

Returns

True or false value.

4.4.1.5 RecInput()

```
Recipe * RecInput (  
    Ingridient * list)
```

4.4.1.6 StrKiller()

```
void StrKiller (  
    char * str)
```

Empties a string

Parameters

<i>str</i>	The string.
------------	-------------

4.5 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/input.h File Reference

```
#include "debugmalloc.h"  
#include "Ingridient.h"  
#include "Recipe.h"
```

Typedefs

- typedef enum [IngInputState](#) [IngInputState](#)
- typedef enum [RecInputState](#) [RecInputState](#)

Enumerations

- enum [IngInputState](#) { [IName](#) , [IType](#) , [CreateIng](#) , [IReturn](#) }
- enum [RecInputState](#) { [RName](#) , [IsFav](#) , [IngCount](#) , [RecipeIngName](#) , [RecipeIngNumber](#) , [RecipeIngType](#) , [CreateRecipeIng](#) , [RecipeCreation](#) , [RReturn](#) }

Functions

- [Ingridient](#) * [IngInput](#) ()
- [Recipe](#) * [RecInput](#) ([Ingridient](#) *list)
- [MeasurementType](#) [IngTypeEnum](#) (char *TypeString)
- char * [IngTypeDenominator](#) ([MeasurementType](#) Mtype)
- bool [IsFavChecker](#) (char *TypeString)
- void [StrKiller](#) (char *str)

4.5.1 Typedef Documentation

4.5.1.1 IngInputState

```
typedef enum IngInputState IngInputState
```

The states used for "Osszetevok.txt" file reading.

4.5.1.2 RecInputState

```
typedef enum RecInputState RecInputState
```

The states used for "Receptek.txt" file reading.

4.5.2 Enumeration Type Documentation

4.5.2.1 IngInputState

```
enum IngInputState
```

The states used for "Osszetevok.txt" file reading.

Enumerator

IName	
IType	
CreateIng	
IReturn	

4.5.2.2 RecInputState

```
enum RecInputState
```

The states used for "Receptek.txt" file reading.

Enumerator

RName	
IsFav	
IngCount	
RecipeIngName	
RecipeIngNumber	
RecipeIngType	
CreateRecipeIng	
RecipeCreation	
RReturn	

4.5.3 Function Documentation

4.5.3.1 IngInput()

```
Ingridient * IngInput ()
```

4.5.3.2 IngTypeDenumerator()

```
char * IngTypeDenumerator (
    MeasurementType Mtype)
```

Converts MeasurementType enum to a string.

Parameters

<i>Mtype</i>	MeasurementType enum that will be converted.
--------------	--

Returns

Address of the string.

4.5.3.3 IngTypeEnumerator()

```
MeasurementType IngTypeEnumerator (
    char * TypeString)
```

Converts a string to a MeasurementType enum.

Parameters

<i>TypeString</i>	The that will be converted
-------------------	----------------------------

Returns

MeasurementType.

4.5.3.4 IsFavChecker()

```
bool IsFavChecker (
    char * TypeString)
```

Converts a string to a true or false value.

Parameters

<i>TypeString</i>	The string that will be converted.
-------------------	------------------------------------

Returns

True or false value.

4.5.3.5 RecInput()

```
Recipe * RecInput (
    Ingridient * list)
```

4.5.3.6 StrKiller()

```
void StrKiller (
    char * str)
```

Empties a string

Parameters

<i>str</i>	The string.
------------	-------------

4.6 input.h

[Go to the documentation of this file.](#)

```
00001 //
00002 // Created by hejag on 23/11/2024.
00003 //
00004
00005 #ifndef INPUT_H
00006 #define INPUT_H
00007 #include "debugmalloc.h"
00008 #include "Ingridient.h"
00009 #include "Recipe.h"
00010
00011 Ingridient *IngInput();
00012
00013 Recipe *RecInput(Ingridient *list);
00014
00015 typedef enum IngInputState {
00016     IName,
00017     IType,
00018     CreateIng,
00019     IReturn
00020 } IngInputState;
00021
00022 typedef enum RecInputState {
00023     RName,
00024     IsFav,
00025     IngCount,
00026     RecipeIngName,
00027     RecipeIngNumber,
00028     RecipeIngType,
00029     CreateRecipeIng,
00030     RecipeCreation,
00031     RReturn
00032 } RecInputState;
00033
00034 MeasurementType IngTypeEnumrator(char *TypeString);
00035
00036 char *IngTypeDenumerator(MeasurementType Mtype);
00037
00038 bool IsFavChecker(char *TypeString);
00039
00040 void StrKiller(char *str);
00041
00042 #endif //INPUT_H
```

4.7 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/main.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "Ingridient.h"
#include "input.h"
#include "Menu.h"
#include "debugmalloc.h"
```

Functions

- int [main](#) (void)

4.7.1 Function Documentation

4.7.1.1 main()

```
int main (
    void )
```

4.8 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Menu.c File Reference

```
#include "Menu.h"
#include <stdio.h>
#include "input.h"
#include "debugmalloc.h"
```

Functions

- void [LogoPrinter](#) ()
- int [MainMenu](#) ([Ingridient](#) **IngListPtr, [Recipe](#) **RecipeList)
- void [SearchMenu](#) ([Ingridient](#) **IngList, [Recipe](#) **RecipeList)
- void [RecipeMenu](#) ([Ingridient](#) **IngList, [Recipe](#) **RecipeList)
- void [IngManageMenu](#) ([Ingridient](#) **IngList, [Recipe](#) **RecipeList)
- void [FavMenu](#) ([Recipe](#) **RecipeList)
- void [RandomRecipe](#) ([Recipe](#) *RecipeList)

4.8.1 Function Documentation

4.8.1.1 FavMenu()

```
void FavMenu (
    Recipe ** RecipeList)
```

The menu for modifying the favorite parameter of the recipe stuct.

Parameters

<i>RecipeList</i>	The address of the first element of the Recipe linked list.
-------------------	---

4.8.1.2 IngManageMenu()

```
void IngManageMenu (  
    Ingridient ** IngList,  
    Recipe ** RecipeList)
```

[Ingridient](#) menu for modifying ingridients.

Parameters

<i>IngList</i>	The address of the first element of the Ingridient linked list.
<i>RecipeList</i>	The address of the first element of the Recipe linked list.

4.8.1.3 LogoPrinter()

```
void LogoPrinter ()
```

Prints out the logo.

4.8.1.4 MainMenu()

```
int MainMenu (  
    Ingridient ** IngListPtr,  
    Recipe ** RecipeList)
```

This is the main menu function this from this function the user can access the other menus.

Parameters

<i>IngListPtr</i>	The address of the first element of the Ingridient linked list.
<i>RecipeList</i>	The address of the first element of the Recipe linked list.

Returns

If the user exits the program with the exit option the return will be 0.

4.8.1.5 RandomRecipe()

```
void RandomRecipe (  
    Recipe * RecipeList)
```

This function will print out a random recipe.

Parameters

<i>RecipeList</i>	The address of the first element of the Recipe linked list.
-------------------	---

4.8.1.6 RecipeMenu()

```
void RecipeMenu (  
    Ingridient ** IngList,  
    Recipe ** RecipeList)
```

[Recipe](#) menu for modifying recipes.

Parameters

<i>IngList</i>	The address of the first element of the Ingridient linked list.
<i>RecipeList</i>	The address of the first element of the Recipe linked list.

4.8.1.7 SearchMenu()

```
void SearchMenu (  
    Ingridient ** IngList,  
    Recipe ** RecipeList)
```

The [Recipe](#) search menu.

Parameters

<i>IngList</i>	The address of the first element of the Ingridient linked list.
<i>RecipeList</i>	The address of the first element of the Recipe linked list.

4.9 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Menu.h File Reference

```
#include "Recipe.h"  
#include "debugmalloc.h"
```

Functions

- void [LogoPrinter](#) ()
- int [MainMenu](#) ([Ingridient](#) **IngListPtr, [Recipe](#) **RecipeList)
- void [SearchMenu](#) ([Ingridient](#) **IngList, [Recipe](#) **RecipeList)
- void [RecipeMenu](#) ([Ingridient](#) **IngList, [Recipe](#) **RecipeList)
- void [IngManageMenu](#) ([Ingridient](#) **IngList, [Recipe](#) **RecipeList)
- void [FavMenu](#) ([Recipe](#) **RecipeList)
- void [RandomRecipe](#) ([Recipe](#) *RecipeList)

4.9.1 Function Documentation

4.9.1.1 FavMenu()

```
void FavMenu (
    Recipe ** RecipeList)
```

The menu for modifying the favorite parameter of the recipe stuct.

Parameters

<i>RecipeList</i>	The address of the first element of the Recipe linked list.
-------------------	---

4.9.1.2 IngManageMenu()

```
void IngManageMenu (
    Ingridient ** IngList,
    Recipe ** RecipeList)
```

[Ingridient](#) menu for modifying ingridients.

Parameters

<i>IngList</i>	The address of the first element of the Ingridient linked list.
<i>RecipeList</i>	The address of the first element of the Recipe linked list.

4.9.1.3 LogoPrinter()

```
void LogoPrinter ()
```

Prints out the logo.

4.9.1.4 MainMenu()

```
int MainMenu (
    Ingridient ** IngListPtr,
    Recipe ** RecipeList)
```

This is the main menu function this from this function the user can access the other menus.

Parameters

<i>IngListPtr</i>	The address of the first element of the Ingridient linked list.
<i>RecipeList</i>	The address of the first element of the Recipe linked list.

Returns

If the user exits the program with the exit option the return will be 0.

4.9.1.5 RandomRecipe()

```
void RandomRecipe (
    Recipe * RecipeList)
```

This function will print out a random recipe.

Parameters

<i>RecipeList</i>	The address of the first element of the Recipe linked list.
-------------------	---

4.9.1.6 RecipeMenu()

```
void RecipeMenu (  
    Ingridient ** IngList,  
    Recipe ** RecipeList)
```

[Recipe](#) menu for modifying recipes.

Parameters

<i>IngList</i>	The address of the first element of the Ingridient linked list.
<i>RecipeList</i>	The address of the first element of the Recipe linked list.

4.9.1.7 SearchMenu()

```
void SearchMenu (  
    Ingridient ** IngList,  
    Recipe ** RecipeList)
```

The [Recipe](#) search menu.

Parameters

<i>IngList</i>	The address of the first element of the Ingridient linked list.
<i>RecipeList</i>	The address of the first element of the Recipe linked list.

4.10 Menu.h

[Go to the documentation of this file.](#)

```
00001 //  
00002 // Created by hejag on 24/11/2024.  
00003 //  
00004  
00005 #ifndef MENU_H  
00006 #define MENU_H  
00007 #include "Recipe.h"  
00008 #include "debugmalloc.h"  
00009  
00013 void LogoPrinter();  
00014  
00021 int MainMenu(Ingridient** IngListPtr, Recipe** RecipeList);  
00022  
00028 void SearchMenu(Ingridient** IngList, Recipe** RecipeList);  
00029  
00035 void RecipeMenu(Ingridient** IngList, Recipe** RecipeList);  
00036  
00042 void IngManageMenu(Ingridient** IngList, Recipe** RecipeList);  
00043  
00048 void FavMenu(Recipe** RecipeList);  
00049  
00054 void RandomRecipe(Recipe* RecipeList);  
00055  
00056  
00057  
00058  
00059 #endif //MENU_H
```

4.11 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Recipe.c File Reference

```
#include "Recipe.h"
#include <stdio.h>
#include "input.h"
#include "Ingridient.h"
#include "debugmalloc.h"
```

Functions

- [Recipe](#) * [CreateRecipe](#) (char *Name, bool blsFav)
- void [AssignIngridients](#) ([Recipe](#) *Rec, [RecipeIngridient](#) *RecIngridient)
- void [AddRecipeTo](#) ([Recipe](#) *list, [Recipe](#) *ToAdd)
- void [AddOrCreateRecipe](#) ([Recipe](#) **list, [Recipe](#) *ToAdd)
- void [DeleteRecipe](#) ([Recipe](#) **listPtr, [Recipe](#) *ToDelete)
- [Recipe](#) * [FindRecipeByName](#) ([Recipe](#) *list, char *RecipeName)
- void [RecipePrint](#) ([Recipe](#) *PRecipe)
- void [PrintRecipeByIng](#) ([Recipe](#) *list)
- void [PrintFavRecipes](#) ([Recipe](#) *list)
- void [SetFavorite](#) ([Recipe](#) *list, char *SetFavName, bool isFav)
- void [PrintAllRecipes](#) ([Recipe](#) *list)
- void [AddRecipeFromIO](#) ([Ingridient](#) *IngList, [Recipe](#) *RecList)
- bool [AddRecipeFromIOIsFav](#) (char *IsFav)
- void [DeleteAllRecipe](#) ([Recipe](#) *list)
- int [RecipeListLen](#) ([Recipe](#) *list)
- char * [BoolFavConverter](#) (bool IsFav)

4.11.1 Function Documentation

4.11.1.1 AddOrCreateRecipe()

```
void AddOrCreateRecipe (
    Recipe ** list,
    Recipe * ToAdd)
```

This function is not used in the program.

Parameters

<i>list</i>	
<i>ToAdd</i>	

4.11.1.2 AddRecipeFromIO()

```
void AddRecipeFromIO (
    Ingridient * IngList,
    Recipe * RecList)
```

This will add a new recipe to the [Recipe](#) linked list. This function gets input from the user.

Parameters

<i>IngList</i>	The address of the first element of the recipe linked list.
<i>RecList</i>	The address of the first element of the ingridient linked list.

4.11.1.3 AddRecipeFromIOIsFav()

```
bool AddRecipeFromIOIsFav (  
    char * IsFav)
```

Gets a string and determines that is it a true or false value when converted to a boolean.

Parameters

<i>IsFav</i>	The string.
--------------	-------------

Returns

The boolean value.

4.11.1.4 AddRecipeTo()

```
void AddRecipeTo (  
    Recipe * list,  
    Recipe * ToAdd)
```

Adds a [Recipe](#) to the [Recipe](#) linked list.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
<i>ToAdd</i>	The address of the Recipe that will be added to the list.

4.11.1.5 AssignIngridients()

```
void AssignIngridients (  
    Recipe * Rec,  
    RecipeIngridient * RecIngridient)
```

Assigns the [RecipeIngridient](#) linked list to and elemnet of [Recipe](#) linkec list.

Parameters

<i>Rec</i>	An element of the Recipe linked list.
<i>RecIngridient</i>	The first element of the RecinpeIngridient lined list.

4.11.1.6 BoolFavConverter()

```
char * BoolFavConverter (  
    bool IsFav)
```

Converts bool to char value.

Parameters

<i>IsFav</i>	bool to be converted.
--------------	-----------------------

Returns

char pointer (string).

4.11.1.7 CreateRecipe()

```
Recipe * CreateRecipe (  
    char * Name,  
    bool bIsFav)
```

This function creates a new recipe.

Parameters

<i>Name</i>	The name of the recipe to create.
<i>bIsFav</i>	Whether to mark this recipe or not.

Returns

Newly created recipe.

4.11.1.8 DeleteAllRecipe()

```
void DeleteAllRecipe (  
    Recipe * list)
```

Deletes all [Recipe](#) from the [Recipe](#) linked list.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

4.11.1.9 DeleteRecipe()

```
void DeleteRecipe (  
    Recipe ** listPtr,  
    Recipe * ToDelete)
```

Deletes an element of the [Recipe](#) linked list specified by the ToDelete pointer.

Parameters

<i>listPtr</i>	The address of the first element of the recipe linked list.
<i>ToDelete</i>	The address of the element of the recipe linked list that will be deleted.

4.11.1.10 FindRecipeByName()

```
Recipe * FindRecipeByName (  
    Recipe * list,  
    char * RecipeName)
```

Finds a recipe by name

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
<i>RecipeName</i>	The name of the recipe.

Returns

Address of the [Recipe](#) specified by name.

4.11.1.11 PrintAllRecipes()

```
void PrintAllRecipes (  
    Recipe * list)
```

Prints all recipes.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

4.11.1.12 PrintFavRecipes()

```
void PrintFavRecipes (  
    Recipe * list)
```

Prints the favorite recipes

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

4.11.1.13 PrintRecipeByIng()

```
void PrintRecipeByIng (  
    Recipe * list)
```

Prints a recipe containing the specified ingredient.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

4.11.1.14 RecipeListLen()

```
int RecipeListLen (  
    Recipe * list)
```

Gets the lenght of the recipe linked list.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

Returns

an intiger value that is the lenght of the list.

4.11.1.15 RecipePrint()

```
void RecipePrint (
    Recipe * PRecipe)
```

Prints the recipe of the specified address.

Parameters

<i>PRecipe</i>	The recipe address.
----------------	---------------------

4.11.1.16 SetFavorite()

```
void SetFavorite (
    Recipe * list,
    char * SetFavName,
    bool isFav)
```

Sets an element of the [Recipe](#) linked list to favorite or not favorite.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
<i>SetFavName</i>	The name of the SelectFavorite Recipe .
<i>isFav</i>	Favorite or not.

4.12 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Recipe.h File Reference

```
#include <stdbool.h>
#include "RecipeIngridient.h"
#include "Ingridient.h"
#include "debugmalloc.h"
```

Data Structures

- struct [Recipe](#)

Typedefs

- typedef struct Recipe [Recipe](#)

Functions

- [Recipe](#) * [CreateRecipe](#) (char *Name, bool blsFav)
- void [AssignIngridients](#) ([Recipe](#) *Rec, [RecipeIngridient](#) *RecIngridient)
- void [AddRecipeTo](#) ([Recipe](#) *list, [Recipe](#) *ToAdd)
- void [AddOrCreateRecipe](#) ([Recipe](#) **list, [Recipe](#) *ToAdd)
- void [DeleteRecipe](#) ([Recipe](#) **listPtr, [Recipe](#) *ToDelete)
- [Recipe](#) * [FindRecipeByName](#) ([Recipe](#) *list, char *RecipeName)
- void [RecipePrint](#) ([Recipe](#) *PRecipe)
- void [PrintRecipeByIng](#) ([Recipe](#) *list)
- void [PrintFavRecipes](#) ([Recipe](#) *list)
- void [SetFavorite](#) ([Recipe](#) *list, char *SetFavName, bool isFav)
- void [PrintAllRecipes](#) ([Recipe](#) *list)
- void [AddRecipeFromIO](#) ([Ingridient](#) *IngList, [Recipe](#) *RecList)
- bool [AddRecipeFromIOIsFav](#) (char *IsFav)
- void [DeleteAllRecipe](#) ([Recipe](#) *list)
- int [RecipeListLen](#) ([Recipe](#) *list)
- char * [BoolFavConverter](#) (bool IsFav)

4.12.1 Typedef Documentation

4.12.1.1 Recipe

```
typedef struct Recipe Recipe
```

The struct that stores recipe values.

4.12.2 Function Documentation

4.12.2.1 AddOrCreateRecipe()

```
void AddOrCreateRecipe (
    Recipe ** list,
    Recipe * ToAdd)
```

This function is not used in the program.

Parameters

<i>list</i>	
<i>ToAdd</i>	

4.12.2.2 AddRecipeFromIO()

```
void AddRecipeFromIO (
    Ingridient * IngList,
    Recipe * RecList)
```

This will add a new recipe to the [Recipe](#) linked list. This function gets input from the user.

Parameters

<i>IngList</i>	The address of the first element of the recipe linked list.
<i>RecList</i>	The address of the first element of the ingridient linked list.

4.12.2.3 AddRecipeFromIOIsFav()

```
bool AddRecipeFromIOIsFav (  
    char * IsFav)
```

Gets a string and determines that is it a true or false value when converted to a boolean.

Parameters

<i>IsFav</i>	The string.
--------------	-------------

Returns

The boolean value.

4.12.2.4 AddRecipeTo()

```
void AddRecipeTo (  
    Recipe * list,  
    Recipe * ToAdd)
```

Adds a [Recipe](#) to the [Recipe](#) linked list.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
<i>ToAdd</i>	The address of the Recipe that will be added to the list.

4.12.2.5 AssignIngridients()

```
void AssignIngridients (  
    Recipe * Rec,  
    RecipeIngridient * RecIngridient)
```

Assigns the [RecipeIngridient](#) linked list to and elemnet of [Recipe](#) linkec list.

Parameters

<i>Rec</i>	An element of the Recipe linked list.
<i>RecIngridient</i>	The first element of the RecinpeIngridient lined list.

4.12.2.6 BoolFavConverter()

```
char * BoolFavConverter (  
    bool IsFav)
```

Converts bool to char value.

Parameters

<i>IsFav</i>	bool to be converted.
--------------	-----------------------

Returns

char pointer (string).

4.12.2.7 CreateRecipe()

```
Recipe * CreateRecipe (  
    char * Name,  
    bool bIsFav)
```

This function creates a new recipe.

Parameters

<i>Name</i>	The name of the recipe to create.
<i>bIsFav</i>	Whether to mark this recipe or not.

Returns

Newly created recipe.

4.12.2.8 DeleteAllRecipe()

```
void DeleteAllRecipe (  
    Recipe * list)
```

Deletes all [Recipe](#) from the [Recipe](#) linked list.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

4.12.2.9 DeleteRecipe()

```
void DeleteRecipe (  
    Recipe ** listPtr,  
    Recipe * ToDelete)
```

Deletes an element of the [Recipe](#) linked list specified by the ToDelete pointer.

Parameters

<i>listPtr</i>	The address of the first element of the recipe linked list.
<i>ToDelete</i>	The address of the element of the recipe linked list that will be deleted.

4.12.2.10 FindRecipeByName()

```
Recipe * FindRecipeByName (  
    Recipe * list,  
    char * RecipeName)
```

Finds a recipe by name

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
<i>RecipeName</i>	The name of the recipe.

Returns

Address of the [Recipe](#) specified by name.

4.12.2.11 PrintAllRecipes()

```
void PrintAllRecipes (  
    Recipe * list)
```

Prints all recipes.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

4.12.2.12 PrintFavRecipes()

```
void PrintFavRecipes (  
    Recipe * list)
```

Prints the favorite recipes

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

4.12.2.13 PrintRecipeByIng()

```
void PrintRecipeByIng (  
    Recipe * list)
```

Prints a recipe containing the specified ingredient.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

4.12.2.14 RecipeListLen()

```
int RecipeListLen (  
    Recipe * list)
```

Gets the lenght of the recipe linked list.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
-------------	---

Returns

an integer value that is the length of the list.

4.12.2.15 RecipePrint()

```
void RecipePrint (
    Recipe * PRecipe)
```

Prints the recipe of the specified address.

Parameters

<i>PRecipe</i>	The recipe address.
----------------	---------------------

4.12.2.16 SetFavorite()

```
void SetFavorite (
    Recipe * list,
    char * SetFavName,
    bool isFav)
```

Sets an element of the [Recipe](#) linked list to favorite or not favorite.

Parameters

<i>list</i>	The address of the first element of the recipe linked list.
<i>SetFavName</i>	The name of the SelectFavorite Recipe .
<i>isFav</i>	Favorite or not.

4.13 Recipe.h

[Go to the documentation of this file.](#)

```
00001 //
00002 // Created by hejag on 23/11/2024.
00003
00004 #ifndef RECIPE_H
00005 #define RECIPE_H
00006 #include <stdbool.h>
00007 #include "RecipeIngridient.h"
00008 #include "Ingridient.h"
00009 #include "debugmalloc.h"
00010
00014 typedef struct Recipe {
00018     char RecName[100];
00022     struct Recipe *NextNode;
00026     RecipeIngridient *RecIng;
00030     bool bIsFavorite;
00031 } Recipe;
```

```

00032
00039 Recipe *CreateRecipe(char *Name, bool bIsFav);
00040
00046 void AssignIngridients(Recipe *Rec, RecipeIngridient *RecIngridient);
00047
00053 void AddRecipeTo(Recipe *list, Recipe *ToAdd);
00054
00060 void AddOrCreateRecipe(Recipe **list, Recipe *ToAdd);
00061
00067 void DeleteRecipe(Recipe **listPtr, Recipe *ToDelete);
00068
00075 Recipe *FindRecipeByName(Recipe *list, char *RecipeName);
00076
00081 void RecipePrint(Recipe *PRecipe);
00082
00087 void PrintRecipeByIng(Recipe* list);
00088
00093 void PrintFavRecipes(Recipe *list);
00094
00101 void SetFavorite(Recipe *list, char *SetFavName, bool isFav);
00102
00107 void PrintAllRecipes(Recipe *list);
00108
00114 void AddRecipeFromIO(Ingridient *IngList, Recipe *RecList);
00115
00121 bool AddRecipeFromIOIsFav(char *IsFav);
00122
00127 void DeleteAllRecipe(Recipe *list);
00128
00134 int RecipeListLen(Recipe *list);
00135
00141 char* BoolFavConverter(bool IsFav);
00142
00143 #endif //RECIPE_H

```

4.14 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/RecipeIngridient.c File Reference

```

#include "RecipeIngridient.h"
#include "debugmalloc.h"

```

Functions

- [RecipeIngridient](#) * CreateRecipeIngridient (int IngAmount, [Ingridient](#) *IngData)
- void AddRITo ([RecipeIngridient](#) *list, [RecipeIngridient](#) *ToAdd)
- void DeleteRI ([RecipeIngridient](#) *list, [RecipeIngridient](#) *ToDelete)
- [RecipeIngridient](#) * FindRecipeIngridientByName ([RecipeIngridient](#) *list, char *IngName)
- void DeleteAllRecIng ([RecipeIngridient](#) *list)

4.14.1 Function Documentation

4.14.1.1 AddRITo()

```

void AddRITo (
    RecipeIngridient * list,
    RecipeIngridient * ToAdd)

```

Adds a [RecipeIngridient](#) struct to the [RecipeIngridient](#) linked list.

Parameters

<i>list</i>	The address of the first element of the RecipeIngridient linked list.
<i>ToAdd</i>	The Pointer of theRecipeIngridient that is added to the RecipeIngridient linked list.

4.14.1.2 CreateRecipeIngridient()

```
RecipeIngridient * CreateRecipeIngridient (  
    int IngAmount,  
    Ingridient * IngData)
```

Creates a RecipeIngridient struct.

Parameters

<i>IngAmount</i>	The amount of the ingridient specified in the ing data.
<i>IngData</i>	Points to the selected Ingridient in the Ingridient linked list.

Returns

Pointer [RecipeIngridient](#) struct.

4.14.1.3 DeleteAllRecIng()

```
void DeleteAllRecIng (  
    RecipeIngridient * list)
```

Deletes all RecipeIngdients from the [RecipeIngridient](#) linked list.

Parameters

<i>list</i>	The address of the first element of the RecipeIngridient linked list.
-------------	---

4.14.1.4 DeleteRI()

```
void DeleteRI (  
    RecipeIngridient * list,  
    RecipeIngridient * ToDelete)
```

Deletes a [RecipeIngridient](#) from the [RecipeIngridient](#) linked list.

Parameters

<i>list</i>	The address of the first element of the RecipeIngridient linked list.
<i>ToDelete</i>	The address of the RecipeIngridient that will be deleted.

4.14.1.5 FindRecipeIngridientByName()

```
RecipeIngridient * FindRecipeIngridientByName (  
    RecipeIngridient * list,  
    char * IngName)
```

Finds a [RecipeIngridient](#) by Name.

Parameters

<i>list</i>	The address of the first element of the RecipeIngridient linked list.
<i>IngName</i>	The name of the searched Ingridient .

Returns

A [RecipeIngridient](#) pointer.

4.15 C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/RecipeIngridient.h File Reference

```
#include "Ingridient.h"
#include "debugmalloc.h"
```

Data Structures

- struct [RecipeIngridient](#)

Typedefs

- typedef struct RecipeIngridient [RecipeIngridient](#)

Functions

- [RecipeIngridient](#) * [CreateRecipeIngridient](#) (int IngAmount, [Ingridient](#) *IngData)
- void [AddRITo](#) ([RecipeIngridient](#) *list, [RecipeIngridient](#) *ToAdd)
- void [DeleteRI](#) ([RecipeIngridient](#) *list, [RecipeIngridient](#) *ToDelete)
- [RecipeIngridient](#) * [FindRecipeIngridientByName](#) ([RecipeIngridient](#) *list, char *IngName)
- void [DeleteAllReclng](#) ([RecipeIngridient](#) *list)

4.15.1 Typedef Documentation

4.15.1.1 RecipeIngridient

```
typedef struct RecipeIngridient RecipeIngridient
```

A wrapper struct that connects the [Ingridient](#) stuct and the [Recipe](#) struct and it stores ingridient amount. The [IngData](#) points to the selected [Ingridient](#). The [NextNode](#) is used to point to the next element of the [RecipeIngridient](#) linked list.

4.15.2 Function Documentation

4.15.2.1 AddRITo()

```
void AddRITo (
    RecipeIngridient * list,
    RecipeIngridient * ToAdd)
```

Adds a [RecipeIngridient](#) struct to the [RecipeIngridient](#) linked list.

Parameters

<i>list</i>	The address of the first element of the RecipeIngridient linked list.
<i>ToAdd</i>	The Pointer of theRecipeIngridient that is added to the RecipeIngridient linked list.

4.15.2.2 CreateRecipeIngridient()

```
RecipeIngridient * CreateRecipeIngridient (  
    int IngAmount,  
    Ingridient * IngData)
```

Creates a RecipeIngridient struct.

Parameters

<i>IngAmount</i>	The amount of the ingridient specified in the ing data.
<i>IngData</i>	Points to the selected Ingridient in the Ingridient linked list.

Returns

Pointer [RecipeIngridient](#) struct.

4.15.2.3 DeleteAllRecIng()

```
void DeleteAllRecIng (  
    RecipeIngridient * list)
```

Deletes all RecipeIngdients from the [RecipeIngridient](#) linked list.

Parameters

<i>list</i>	The address of the first element of the RecipeIngridient linked list.
-------------	---

4.15.2.4 DeleteRI()

```
void DeleteRI (  
    RecipeIngridient * list,  
    RecipeIngridient * ToDelete)
```

Deletes a [RecipeIngridient](#) from the [RecipeIngridient](#) linked list.

Parameters

<i>list</i>	The address of the first element of the RecipeIngridient linked list.
<i>ToDelete</i>	The address of the RecipeIngridient that will be deleted.

4.15.2.5 FindRecipeIngridientByName()

```
RecipeIngridient * FindRecipeIngridientByName (  
    RecipeIngridient * list,  
    char * IngName)
```

Finds a [RecipeIngridient](#) by Name.

Parameters

<i>list</i>	The address of the first element of the RecipeIngridient linked list.
<i>IngName</i>	The name of the searched Ingridient .

Returns

A [RecipeIngridient](#) pointer.

4.16 RecipeIngridient.h

[Go to the documentation of this file.](#)

```
00001 //
00002 // Created by hejag on 24/11/2024.
00003
00004 #ifndef RECIPEINGRIDIENT_H
00005 #define RECIPEINGRIDIENT_H
00006 #include "Ingridient.h"
00007 #include "debugmalloc.h"
00011 typedef struct RecipeIngridient {
00015     int IngAmount;
00019     Ingridient *IngData;
00023     struct RecipeIngridient *NextNode;
00024 } RecipeIngridient;
00025
00032 RecipeIngridient *CreateRecipeIngridient(int IngAmount, Ingridient *IngData);
00033
00039 void AddRITo(RecipeIngridient *list, RecipeIngridient *ToAdd);
00040
00046 void DeleteRI(RecipeIngridient *list, RecipeIngridient *ToDelete);
00047
00054 RecipeIngridient *FindRecipeIngridientByName(RecipeIngridient *list, char *IngName);
00055
00060 void DeleteAllRecIng(RecipeIngridient *list);
00061
00062 //void PrintIngName(Ingridient *list, int IngridientIndex);
00063
00064 #endif //RECIPEINGRIDIENT_H
```

Index

AddIngridientTo
 Ingridient.c, [9](#)
 Ingridient.h, [13](#)
AddOrCreatelNg
 Ingridient.c, [9](#)
 Ingridient.h, [13](#)
AddOrCreateRecipe
 Recipe.c, [26](#)
 Recipe.h, [31](#)
AddRecipeFromIO
 Recipe.c, [26](#)
 Recipe.h, [31](#)
AddRecipeFromIOIsFav
 Recipe.c, [27](#)
 Recipe.h, [32](#)
AddRecipeTo
 Recipe.c, [27](#)
 Recipe.h, [32](#)
AddRITo
 RecipeIngridient.c, [36](#)
 RecipeIngridient.h, [38](#)
AssignIngridients
 Recipe.c, [27](#)
 Recipe.h, [32](#)

blsFavorite
 Recipe, [6](#)
BoolFavConverter
 Recipe.c, [27](#)
 Recipe.h, [32](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Ingridient.c,
[9](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Ingridient.h,
[12](#), [15](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/input.c,
[16](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/input.h,
[17](#), [20](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/main.c,
[21](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Menu.c,
[21](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Menu.h,
[23](#), [25](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Recipe.c,
[26](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Recipe.h,
[30](#), [35](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Recipe.h,
[36](#)

C:/Users/hejag/Documents/School Projects/ProgAlap01/NHF/NHFSave/Recipe.h,
[38](#), [40](#)

CreateIng
 input.h, [18](#)

CreateIngridient
 Ingridient.c, [10](#)
 Ingridient.h, [13](#)

CreateRecipe
 Recipe.c, [28](#)
 Recipe.h, [33](#)

CreateRecipeIng
 input.h, [19](#)

CreateRecipeIngridient
 RecipeIngridient.c, [37](#)
 RecipeIngridient.h, [39](#)

darab
 Ingridient.h, [13](#)

DeleteAllIng
 Ingridient.c, [10](#)
 Ingridient.h, [13](#)

DeleteAllRecIng
 RecipeIngridient.c, [37](#)
 RecipeIngridient.h, [39](#)

DeleteAllRecipe
 Recipe.c, [28](#)
 Recipe.h, [33](#)

DeleteIngridient
 Ingridient.c, [10](#)
 Ingridient.h, [13](#)

DeleteRecipe
 Recipe.c, [28](#)
 Recipe.h, [33](#)

DeleteRecipeIng
 RecipeIngridient.c, [37](#)
 RecipeIngridient.h, [39](#)

DeleteRecipeIngridient
 RecipeIngridient.c, [37](#)
 RecipeIngridient.h, [39](#)

FindIngridientByName
 Menu.c, [21](#)

FindIngridientByName
 Ingridient.h, [14](#)

FindRecipeByRecipe
 Recipe.c, [28](#)
 Recipe.h, [33](#)

FindRecipeIngridientByName
 RecipeIngridient.c, [37](#)

- RecipeIngredient.h, 39
- IName
 - input.h, 18
- IngAmount
 - RecipeIngredient, 7
- IngCount
 - input.h, 19
- IngData
 - RecipeIngredient, 7
- IngInput
 - input.c, 16
 - input.h, 19
- IngInputState
 - input.h, 18
- IngManageMenu
 - Menu.c, 22
 - Menu.h, 24
- IngName
 - Ingridient, 5
- Ingridient, 5
 - IngName, 5
 - Ingridient.h, 12
 - MType, 5
 - NextNode, 5
- Ingridient.c
 - AddIngridientTo, 9
 - AddOrCreateIng, 9
 - CreateIngridient, 10
 - DeleteAllIng, 10
 - DeleteIngridient, 10
 - FindIngridientByName, 11
 - PrintAllIng, 11
 - PrintIngNameByIndex, 11
- Ingridient.h
 - AddIngridientTo, 13
 - AddOrCreateIng, 13
 - CreateIngridient, 13
 - darab, 13
 - DeleteAllIng, 13
 - DeleteIngridient, 14
 - FindIngridientByName, 14
 - Ingridient, 12
 - kilogramm, 13
 - liter, 13
 - MeasurementType, 12
 - PrintAllIng, 14
 - PrintIngNameByIndex, 15
 - Recipe, 12
- IngTypeDenominator
 - input.c, 16
 - input.h, 19
- IngTypeEnumulator
 - input.c, 16
 - input.h, 19
- input.c
 - IngInput, 16
 - IngTypeDenominator, 16
 - IngTypeEnumulator, 16
- IsFavChecker, 17
- RecInput, 17
- StrKiller, 17
- input.h
 - CreateIng, 18
 - CreateRecipeIng, 19
 - IName, 18
 - IngCount, 19
 - IngInput, 19
 - IngInputState, 18
 - IngTypeDenominator, 19
 - IngTypeEnumulator, 19
 - IReturn, 18
 - IsFav, 19
 - IsFavChecker, 19
 - IType, 18
 - RecInput, 20
 - RecInputState, 18
 - RecipeCreation, 19
 - RecipeIngName, 19
 - RecipeIngNumber, 19
 - RecipeIngType, 19
 - RName, 19
 - RReturn, 19
 - StrKiller, 20
- IReturn
 - input.h, 18
- IsFav
 - input.h, 19
- IsFavChecker
 - input.c, 17
 - input.h, 19
- IType
 - input.h, 18
- kilogramm
 - Ingridient.h, 13
- liter
 - Ingridient.h, 13
- LogoPrinter
 - Menu.c, 22
 - Menu.h, 24
- main
 - main.c, 21
- main.c
 - main, 21
- MainMenu
 - Menu.c, 22
 - Menu.h, 24
- MeasurementType
 - Ingridient.h, 12
- Menu.c
 - FavMenu, 21
 - IngManageMenu, 22
 - LogoPrinter, 22
 - MainMenu, 22
 - RandomRecipe, 22

- RecipeMenu, 23
- SearchMenu, 23
- Menu.h
 - FavMenu, 24
 - IngManageMenu, 24
 - LogoPrinter, 24
 - MainMenu, 24
 - RandomRecipe, 24
 - RecipeMenu, 25
 - SearchMenu, 25
- MType
 - Ingridient, 5
- NextNode
 - Ingridient, 5
 - Recipe, 6
 - RecipeIngridient, 7
- PrintAllIng
 - Ingridient.c, 11
 - Ingridient.h, 14
- PrintAllRecipes
 - Recipe.c, 29
 - Recipe.h, 34
- PrintFavRecipes
 - Recipe.c, 29
 - Recipe.h, 34
- PrintIngNameByIndex
 - Ingridient.c, 11
 - Ingridient.h, 15
- PrintRecipeByIng
 - Recipe.c, 29
 - Recipe.h, 34
- RandomRecipe
 - Menu.c, 22
 - Menu.h, 24
- RecIng
 - Recipe, 6
- RecInput
 - input.c, 17
 - input.h, 20
- RecInputState
 - input.h, 18
- Recipe, 6
 - blsFavorite, 6
 - Ingridient.h, 12
 - NextNode, 6
 - RecIng, 6
 - Recipe.h, 31
 - RecName, 6
- Recipe.c
 - AddOrCreateRecipe, 26
 - AddRecipeFromIO, 26
 - AddRecipeFromIOIsFav, 27
 - AddRecipeTo, 27
 - AssignIngridients, 27
 - BoolFavConverter, 27
 - CreateRecipe, 28
 - DeleteAllRecipe, 28
 - DeleteRecipe, 28
 - FindRecipeByName, 28
 - PrintAllRecipes, 29
 - PrintFavRecipes, 29
 - PrintRecipeByIng, 29
 - RecipeListLen, 29
 - RecipePrint, 30
 - SetFavorite, 30
- Recipe.h
 - AddOrCreateRecipe, 31
 - AddRecipeFromIO, 31
 - AddRecipeFromIOIsFav, 32
 - AddRecipeTo, 32
 - AssignIngridients, 32
 - BoolFavConverter, 32
 - CreateRecipe, 33
 - DeleteAllRecipe, 33
 - DeleteRecipe, 33
 - FindRecipeByName, 33
 - PrintAllRecipes, 34
 - PrintFavRecipes, 34
 - PrintRecipeByIng, 34
 - Recipe, 31
 - RecipeListLen, 34
 - RecipePrint, 35
 - SetFavorite, 35
- RecipeCreation
 - input.h, 19
- RecipeIngName
 - input.h, 19
- RecipeIngNumber
 - input.h, 19
- RecipeIngridient, 7
 - IngAmount, 7
 - IngData, 7
 - NextNode, 7
 - RecipeIngridient.h, 38
- RecipeIngridient.c
 - AddRITo, 36
 - CreateRecipeIngridient, 37
 - DeleteAllRecIng, 37
 - DeleteRI, 37
 - FindRecipeIngridientByName, 37
- RecipeIngridient.h
 - AddRITo, 38
 - CreateRecipeIngridient, 39
 - DeleteAllRecIng, 39
 - DeleteRI, 39
 - FindRecipeIngridientByName, 39
 - RecipeIngridient, 38
- RecipeIngType
 - input.h, 19
- RecipeListLen
 - Recipe.c, 29
 - Recipe.h, 34
- RecipeMenu
 - Menu.c, 23

- Menu.h, [25](#)
- RecipePrint
 - Recipe.c, [30](#)
 - Recipe.h, [35](#)
- RecName
 - Recipe, [6](#)
- RName
 - input.h, [19](#)
- RReturn
 - input.h, [19](#)
- SearchMenu
 - Menu.c, [23](#)
 - Menu.h, [25](#)
- SetFavorite
 - Recipe.c, [30](#)
 - Recipe.h, [35](#)
- StrKiller
 - input.c, [17](#)
 - input.h, [20](#)