Millionairs

Generated by Doxygen 1.8.16

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 leaderboard Struct Reference

`#include <struktury.h>`

Data Fields

- char name [50]
- char how_much_money [50]
- struct leaderboard * pNext

3.1.1 Detailed Description

This struct is created in order to store players names, information about their incomes during game and a pointer to the next player.

3.1.2 Field Documentation

3.1.2.1 how_much_money

`char how_much_money[50]`

3.1.2.2 name

`char name[50]`

3.1.2.3 pNext

```
struct leaderboard* pNext
```

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

- Milionerzy_Projekt_Marek_Kawalski/struktury.h

3.2 question Struct Reference

```
#include <struktury.h>
```

Data Fields

- char question [100]
- char answer_A [100]
- char answer_B [100]
- char answer_C [100]
- char answer_D [100]
- char correct_answer [10]
- int number_of_question
- bool has_it_appeared
- struct question * pNext

3.2.1 Detailed Description

This struct is used so as to have an access to data connected with questions, namely the questions themselves, a,b,c or d variants of plausible answers and finally correct answers. It also contains information if the question has already been used. I used typedef so as not to use a key word "struct" prior to each pHead pointer. It also contains a pointer to the next question.

3.2.2 Field Documentation

3.2.2.1 answer_A

```
char answer_A[100]
```

3.2.2.2 answer_B

```
char answer_B[100]
```

3.2.2.3 answer_C

```
char answer_C[100]
```

3.2.2.4 answer_D

```
char answer_D[100]
```

3.2.2.5 correct_answer

```
char correct_answer[10]
```

3.2.2.6 has_it_appeared

```
bool has_it_appeared
```

3.2.2.7 number_of_question

```
int number_of_question
```

3.2.2.8 pNext

```
struct question* pNext
```

3.2.2.9 question

```
char question[100]
```

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

- Milionerzy_Projekt_Marek_Kawalski/struktury.h

Chapter 4

File Documentation

4.1 Milionerzy_Projekt_Marek_Kawalski/funkcje.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "struktury.h"
#include "funkcje.h"
#include <string.h>
#include <time.h>
```

Macros

- #define _CRT_SECURE_NO_WARNINGS

Functions

- void read_from_file (quest **pHead, char *name_of_file)
- void create_a_single_linked_list_by_pushing_back (quest **pHead, char question[MAX], char answer_A[MAX], char answer_B[MAX], char answer_C[MAX], char answer_D[MAX], char correct_answer[10])
- void print_list (quest *pHead)
- void delete_list (quest **pHead)
- quest * search_question (quest *pHead, int question_number)
- int amount_of_questions (quest *pHead)
- void give_us_random_question (quest **pHead, leaderboard **Head, char **filename)
- int generate_random_number (int number)
- void how_much_money (int k)
- void what_question (int k)
- void display_name ()
- void sub_menu (quest **pHead, leaderboard **Head, char **filename)
- void how_to_play ()
- void about ()
- void game_manu (quest **pHead, leaderboard **Head, char **filename)
- void write_to_file (leaderboard **Head, char **filename, char name[50], char money[50], int minutes, int seconds)
- void create_list_of_players_by_pushing_front (leaderboard **pHead, char name[50], char money[50])

- void wait (int seconds)
- void call_a_friend (quest *pHead, char name[50], int number_of_question)
- void fifty_fifty (quest *pHead, int number_of_question)
- void ask_the_audience (quest *pHead, int number_of_question)
- void delete_list_of_players (leaderboard **pHead)
- void print_list_of_players (leaderboard *pHead)
- void check_guard (quest **pHead)

4.1.1 Macro Definition Documentation

4.1.1.1 _CRT_SECURE_NO_WARNINGS

```
#define _CRT_SECURE_NO_WARNINGS
```

4.1.2 Function Documentation

4.1.2.1 about()

```
void about ( )
```

Simple void function which tells the user the story about original millionairs

Parameters

lack	of parameters

Returns

returns nothing

4.1.2.2 amount_of_questions()

```
int amount_of_questions (
            quest * pHead )
```

Function calculates amount of questions in the list.

Parameters

pHead	pointer to the list

Returns

returns amount of questions

4.1.2.3 ask_the_audience()

```
void ask_the_audience (
            quest * pHead,
            int number_of_question )
```

Function is used as a ask_the_audience lifeline.

Parameters

pHead	pointer to the beginning of the list
number_of_question	it's current's question number

Returns

returns nothing

4.1.2.4 call_a_friend()

```
void call_a_friend (
            quest * pHead,
            char name[50],
            int number_of_question )
```

Function is used as a call a friend lifeline.

Parameters

pHead	pointer to the beginning of the list
name	user's name
number_of_question	it's current's question number

Returns

returns nothing

4.1.2.5 check_guard()

```
void check_guard (
            quest ** pHead )
```

It's an important function which checks whether all of the questions have been used or not. If yes, it automatically turns them all into unused questions and therefore questions can repeat after the limit is exceeded.

Parameters

Head	pointer to the pointer to the beginning of the list (original value)

Returns

returns nothing

4.1.2.6 create_a_single_linked_list_by_pushing_back()

```
void create_a_single_linked_list_by_pushing_back (
            quest ** pHead,
            char question[MAX],
            char answer_A[MAX],
            char answer_B[MAX],
            char answer_C[MAX],
            char answer_D[MAX],
            char correct_answer[10] )
```

This function is used to create a single linked list by pushing back new elements. It's used in collaboration with read_from_file function

Parameters

pHead	Pointer to a pointer to the beginning of the list
question[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
answerA[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
answerB[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
answerC[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
answerD[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
correct_answer[10]	it's a string

See also

{read_from_file}

Returns

returns nothing

4.1.2.7 create_list_of_players_by_pushing_front()

```
void create_list_of_players_by_pushing_front (
            leaderboard ** pHead,
            char name[50],
            char money[50] )
```

Function creates list of names by pushing front. It could have been implemented in a similar way to create_a_↩ single_linked_list_by_pushing_back but so as to further practice my programming skills I choose another way.

Parameters

Head	pointer to the pointer to the beginning of the second list (also original value)
name	it's the players name
money	it's how much has the player won

4.1.2.8 delete_list()

```
void delete_list (
            quest ** pHead )
```

This function deletes list of questions and answers so as not to have memory leaks

Parameters

pHead	Pointer to a pointer to the beginning of the list

Returns

returns nothing

4.1.2.9 delete_list_of_players()

```
void delete_list_of_players (
            leaderboard ** pHead )
```

Function deletes list of players.

Parameters

Head	pointer to the pointer to the beginning of the second list (original value)

Returns

returns nothing

4.1.2.10 display_name()

```
void display_name ( )
```

Function that using '$' characters displays "millionairs".

Parameters

no	parameters

Returns

returns nothing

4.1.2.11 fifty_fifty()

```
void fifty_fifty (
            quest * pHead,
            int number_of_question )
```

Function is used as a fifty_fifty lifeline.

Parameters

pHead	pointer to the beginning of the list

Returns

returns nothing

4.1.2.12 game_manu()

```
void game_manu (
            quest ** pHead,
            leaderboard ** Head,
            char ** filename )
```

Function which gathers various captions and couple of other functions.

Parameters

pHead	pointer to the pointer to the beginning of the list (working on original value)
Head	pointer to the pointer to the beginning of the second list (also original value)
filename	original file

Returns

returns nothing

4.1.2.13 generate_random_number()

```
int generate_random_number (
            int number )
```

Simple function which generates integer random number.

Parameters

number	it's the maximum value that can be generated

Returns

returns random number

4.1.2.14 give_us_random_question()

```
void give_us_random_question (
            quest ** pHead,
            leaderboard ** Head,
            char ** filename )
```

This is the foremost function in which most of other functions is executed.

Parameters

pHead	pointer to the pointer to the beginning of the list (working on original value)
Head	pointer to the pointer to the beginning of the second list (also original value)

See also

{search_question}

{amount_of_questions}

{generate_random_number}

{how_much_money}

{what_question}

{call_a_friend}

{fifty_fifty}

{ask_the_audience}

Returns

returns nothing

4.1.2.15 how_much_money()

```
void how_much_money (
            int k )
```

Void function which assigment is to display the amount of money the player can win.

Parameters

k	parameter which is essential to know what amount to money display with each question.

Returns

returns nothing

4.1.2.16 how_to_play()

```
void how_to_play ( )
```

Simple void function which tells the user how to play.

Parameters

lack	of parameters

Returns

returns nothing

4.1.2.17 print_list()

```
void print_list (
            quest * pHead )
```

Additional function which was used to check if both creating and deleting the list works properly

Parameters

pHead	pointer to the beginning of the list

Returns

returns nothing

4.1.2.18 print_list_of_players()

```
void print_list_of_players (
            leaderboard * pHead )
```

A test function which prints list's of players content.

Parameters

pHead	pointer to the beginning of the list

Returns

returns nothing

4.1.2.19 read_from_file()

```
void read_from_file (
            quest ** pHead,
            char * name_of_file )
```

Function is used to read line by line data from file using gets function. All data is stored in one directional list which is created by "create_a_single_linked_list_by_pushing_back" function".

Parameters

pHead	Pointer to a pointer to the beginning of the list
name_of_file	name of the file where there are questions and answers

See also

{create_a_single_linked_list_by_pushing_back}

Returns

returns nothing

4.1.2.20 search_question()

```
quest* search_question (
            quest * pHead,
            int question_number )
```

This function searches question by the number of question and returns pointer to that question. Its used in give_↩ us_random_question function.

Parameters

pHead	pointer to the beginning of the list
question_number	int parameter

Returns

returns pointer to to the question

4.1.2.21 sub_menu()

```
void sub_menu (
            quest ** pHead,
            leaderboard ** Head,
            char ** filename )
```

Vital function which gives the player a choice of what to do and then executes adequate functions.

Parameters

pHead	pointer to the pointer to the beginning of the list (working on original value)
Head	pointer to the pointer to the beginning of the second list (also original value)
filename	original file

Returns

returns nothing

4.1.2.22 wait()

```
void wait (
            int seconds )
```

Function reads system time and makes the programm wait by the time specified by parameter.

Parameters

seconds	how long the programm is held waiting

Returns

returns nothing

4.1.2.23 what_question()

```
void what_question (
            int k )
```

Void function which is similar to "how_much_money" function. The only diffrence is that it's displayed every time disregard whether the answer had been correct or not.

Parameters

k	parameter which is crucial to know what amount to money display with each question.

Returns

returns nothing

4.1.2.24 write_to_file()

```
void write_to_file (
            leaderboard ** pHead,
            char ** filename,
            char name[50],
            char money[50],
            int minutes,
            int seconds )
```

Function is used to open file in append mode.

Parameters

pHead	pointer to the pointer to the beginning of the second list (also original value)

Parameters

filename	original file
name	it's the players name
money	it's how much has the player won
minutes	it's how long has the game lasted
seconds	it's how long has the game lasted

Returns

returns nothing

4.2 Milionerzy_Projekt_Marek_Kawalski/funkcje.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "struktury.h"
```

Macros

- #define _CRT_SECURE_NO_WARNINGS
- #define MAX 100

Functions

- void read_from_file (quest **pHead, char *name_of_file)
- void print_list (quest *pHead)
- void create_a_single_linked_list_by_pushing_back (quest **pHead, char question[MAX], char answer_A[MAX], char answer_B[MAX], char answer_C[MAX], char answer_D[MAX], char correct_answer[10])
- void delete_list (quest **pHead)
- quest * search_question (quest *pHead, int question_number)
- int amount_of_questions (quest *pHead)
- void give_us_random_question (quest **pHead, leaderboard **Head, char **filename)
- int generate_random_number (int number)
- void how_much_money (int k)
- void what_question (int k)
- void display_name ()
- void sub_menu (quest **pHead, leaderboard **Head, char **filename)
- void how_to_play ()
- void about ()
- void game_manu (quest **pHead, leaderboard **Head, char **filename)
- void write_to_file (leaderboard **pHead, char **filename, char name[50], char money[50], int minutes, int seconds)
- void create_list_of_players_by_pushing_front (leaderboard **pHead, char name[50], char money[50])
- void wait (int seconds)
- void call_a_friend (quest *pHead, char name[50], int number_of_question)
- void fifty_fifty (quest *pHead, int number_of_question)
- void ask_the_audience (quest *pHead, int number_of_question)
- void delete_list_of_players (leaderboard **pHead)
- void print_list_of_players (leaderboard *pHead)
- void check_guard (quest **pHead)

4.2.1 Macro Definition Documentation

4.2.1.1 _CRT_SECURE_NO_WARNINGS

```
#define _CRT_SECURE_NO_WARNINGS
```

This macro removes secure warnings.

4.2.1.2 MAX

```
#define MAX 100
```

This macro defines maximum value which is used in couple of functions.

See also

{read_from_file}

{create_a_single_linked_list_by_pushing_back}

4.2.2 Function Documentation

4.2.2.1 about()

```
void about ( )
```

Simple void function which tells the user the story about original millionairs

Parameters

lack	of parameters

Returns

returns nothing

4.2.2.2 amount_of_questions()

```
int amount_of_questions (
            quest * pHead )
```

Function calculates amount of questions in the list.

Parameters

pHead	pointer to the list

Returns

returns amount of questions

4.2.2.3 ask_the_audience()

```
void ask_the_audience (
            quest * pHead,
            int number_of_question )
```

Function is used as a ask_the_audience lifeline.

Parameters

pHead	pointer to the beginning of the list
number_of_question	it's current's question number

Returns

returns nothing

4.2.2.4 call_a_friend()

```
void call_a_friend (
            quest * pHead,
            char name[50],
            int number_of_question )
```

Function is used as a call a friend lifeline.

Parameters

pHead	pointer to the beginning of the list
name	user's name
number_of_question	it's current's question number

Returns

returns nothing

4.2.2.5 check_guard()

```
void check_guard (
            quest ** pHead )
```

It's an important function which checks whether all of the questions have been used or not. If yes, it automatically turns them all into unused questions and therefore questions can repeat after the limit is exceeded.

Parameters

Head	pointer to the pointer to the beginning of the list (original value)

Returns

returns nothing

4.2.2.6 create_a_single_linked_list_by_pushing_back()

```
void create_a_single_linked_list_by_pushing_back (
            quest ** pHead,
            char question[MAX],
            char answer_A[MAX],
            char answer_B[MAX],
            char answer_C[MAX],
            char answer_D[MAX],
            char correct_answer[10] )
```

This function is used to create a single linked list by pushing back new elements. It's used in collaboration with read_from_file function

Parameters

pHead	Pointer to a pointer to the beginning of the list
question[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
answerA[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
answerB[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
answerC[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
answerD[MAX]	it's a string which size is defined at the beginning of "funkcje.h" file
correct_answer[10]	it's a string

See also

{read_from_file}

Returns

returns nothing

4.2.2.7 create_list_of_players_by_pushing_front()

```
void create_list_of_players_by_pushing_front (
            leaderboard ** pHead,
            char name[50],
            char money[50] )
```

Function creates list of names by pushing front. It could have been implemented in a similar way to create_a_↩ single_linked_list_by_pushing_back but so as to further practice my programming skills I choose another way.

Parameters

Head	pointer to the pointer to the beginning of the second list (also original value)
name	it's the players name
money	it's how much has the player won

4.2.2.8 delete_list()

```
void delete_list (
            quest ** pHead )
```

This function deletes list of questions and answers so as not to have memory leaks

Parameters

pHead	Pointer to a pointer to the beginning of the list

Returns

returns nothing

4.2.2.9 delete_list_of_players()

```
void delete_list_of_players (
            leaderboard ** pHead )
```

Function deletes list of players.

Parameters

Head	pointer to the pointer to the beginning of the second list (original value)

Returns

returns nothing

4.2.2.10 display_name()

```
void display_name ( )
```

Function that using '$' characters displays "millionairs".

Parameters

no	parameters

Returns

returns nothing

4.2.2.11 fifty_fifty()

```
void fifty_fifty (
            quest * pHead,
            int number_of_question )
```

Function is used as a fifty_fifty lifeline.

Parameters

pHead	pointer to the beginning of the list

Returns

returns nothing

4.2.2.12 game_manu()

```
void game_manu (
            quest ** pHead,
            leaderboard ** Head,
            char ** filename )
```

Function which gathers various captions and couple of other functions.

Parameters

pHead	pointer to the pointer to the beginning of the list (working on original value)
Head	pointer to the pointer to the beginning of the second list (also original value)
filename	original file

Returns

returns nothing

4.2.2.13 generate_random_number()

```
int generate_random_number (
            int number )
```

Simple function which generates integer random number.

Parameters

number	it's the maximum value that can be generated

Returns

returns random number

4.2.2.14 give_us_random_question()

```
void give_us_random_question (
            quest ** pHead,
            leaderboard ** Head,
            char ** filename )
```

This is the foremost function in which most of other functions is executed.

Parameters

pHead	pointer to the pointer to the beginning of the list (working on original value)
Head	pointer to the pointer to the beginning of the second list (also original value)

See also

{search_question}

{amount_of_questions}

{generate_random_number}

{how_much_money}

{what_question}

{call_a_friend}

{fifty_fifty}

{ask_the_audience}

Returns

returns nothing

4.2.2.15 how_much_money()

```
void how_much_money (
            int k )
```

Void function which assigment is to display the amount of money the player can win.

Parameters

k	parameter which is essential to know what amount to money display with each question.

Returns

returns nothing

4.2.2.16 how_to_play()

```
void how_to_play ( )
```

Simple void function which tells the user how to play.

Parameters

lack	of parameters

Returns

returns nothing

4.2.2.17 print_list()

```
void print_list (
            quest * pHead )
```

Additional function which was used to check if both creating and deleting the list works properly

Parameters

pHead	pointer to the beginning of the list

Returns

returns nothing

4.2.2.18 print_list_of_players()

```
void print_list_of_players (
            leaderboard * pHead )
```

A test function which prints list's of players content.

Parameters

pHead	pointer to the beginning of the list

Returns

returns nothing

4.2.2.19 read_from_file()

```
void read_from_file (
            quest ** pHead,
            char * name_of_file )
```

Function is used to read line by line data from file using gets function. All data is stored in one directional list which is created by "create_a_single_linked_list_by_pushing_back" function".

Parameters

pHead	Pointer to a pointer to the beginning of the list
name_of_file	name of the file where there are questions and answers

See also

{create_a_single_linked_list_by_pushing_back}

Returns

returns nothing

4.2.2.20 search_question()

```
quest* search_question (
            quest * pHead,
            int question_number )
```

This function searches question by the number of question and returns pointer to that question. Its used in give_↩ us_random_question function.

Parameters

pHead	pointer to the beginning of the list
question_number	int parameter

Returns

returns pointer to to the question

4.2.2.21 sub_menu()

```
void sub_menu (
            quest ** pHead,
            leaderboard ** Head,
            char ** filename )
```

Vital function which gives the player a choice of what to do and then executes adequate functions.

Parameters

pHead	pointer to the pointer to the beginning of the list (working on original value)
Head	pointer to the pointer to the beginning of the second list (also original value)
filename	original file

Returns

returns nothing

4.2.2.22 wait()

```
void wait (
            int seconds )
```

Function reads system time and makes the programm wait by the time specified by parameter.

Parameters

seconds	how long the programm is held waiting

Returns

returns nothing

4.2.2.23 what_question()

```
void what_question (
            int k )
```

Void function which is similar to "how_much_money" function. The only diffrence is that it's displayed every time disregard whether the answer had been correct or not.

Parameters

k	parameter which is crucial to know what amount to money display with each question.

Returns

returns nothing

4.2.2.24 write_to_file()

```
void write_to_file (
            leaderboard ** pHead,
            char ** filename,
            char name[50],
            char money[50],
            int minutes,
            int seconds )
```

Function is used to open file in append mode.

Parameters

pHead	pointer to the pointer to the beginning of the second list (also original value)

Parameters

filename	original file
name	it's the players name
money	it's how much has the player won
minutes	it's how long has the game lasted
seconds	it's how long has the game lasted

Returns

returns nothing

4.3 Milionerzy_Projekt_Marek_Kawalski/Milionerzy_Projekt_Marek_Kawalski.c File Reference

```
#include <stdlib.h>
#include <crtdbg.h>
#include <stdio.h>
#include "funkcje.h"
#include "struktury.h"
```

Functions

- int main ()

4.3.1 Function Documentation

4.3.1.1 main()

```
int main ( )
```

4.4 Milionerzy_Projekt_Marek_Kawalski/struktury.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
```

Data Structures

- struct question
- struct leaderboard

Macros

- #define _CRT_SECURE_NO_WARNINGS

Typedefs

- typedef struct question quest
- typedef struct leaderboard leaderboard

Enumerations

- enum bool { TRUE = 1, FALSE = 0 }

4.4.1 Macro Definition Documentation

4.4.1.1 _CRT_SECURE_NO_WARNINGS

```
#define _CRT_SECURE_NO_WARNINGS
```

This macro removes secure warnings.

4.4.2 Typedef Documentation

4.4.2.1 leaderboard

```
typedef struct leaderboard leaderboard
```

This struct is created in order to store players names, information about their incomes during game and a pointer to the next player.

4.4.2.2 quest

```
typedef struct question quest
```

This struct is used so as to have an access to data connected with questions, namely the questions themselves, a,b,c or d variants of plausible answers and finally correct answers. It also contains information if the question has already been used. I used typedef so as not to use a key word "struct" prior to each pHead pointer. It also contains a pointer to the next question.

4.4.3 Enumeration Type Documentation

4.4.3.1 bool

```
enum bool
```

Enumeration which defines bool type. Bool has proved to be very useful while checking if the questions have appeared.

Enumerator

TRUE	
FALSE	

Index