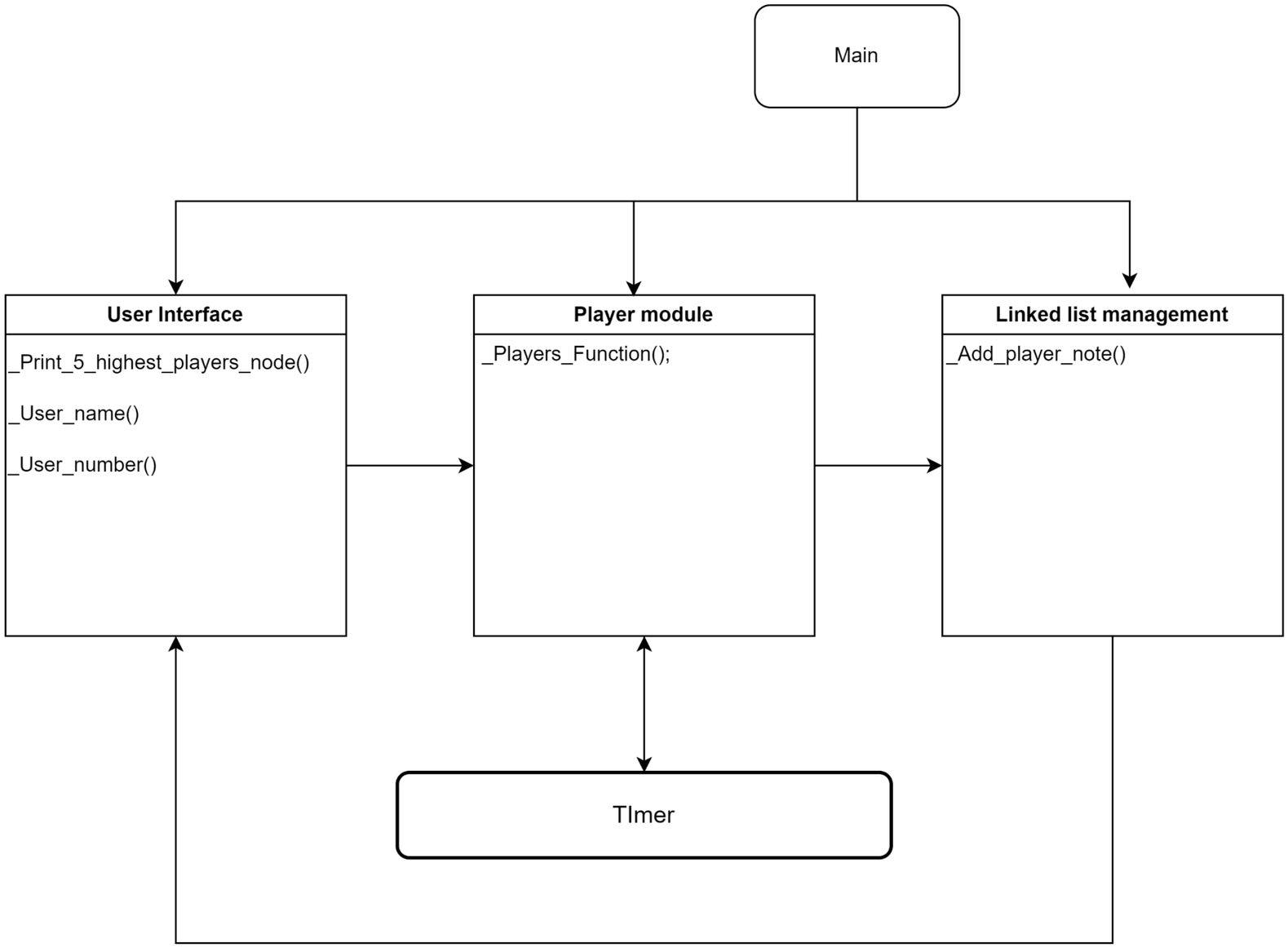
Architecture design



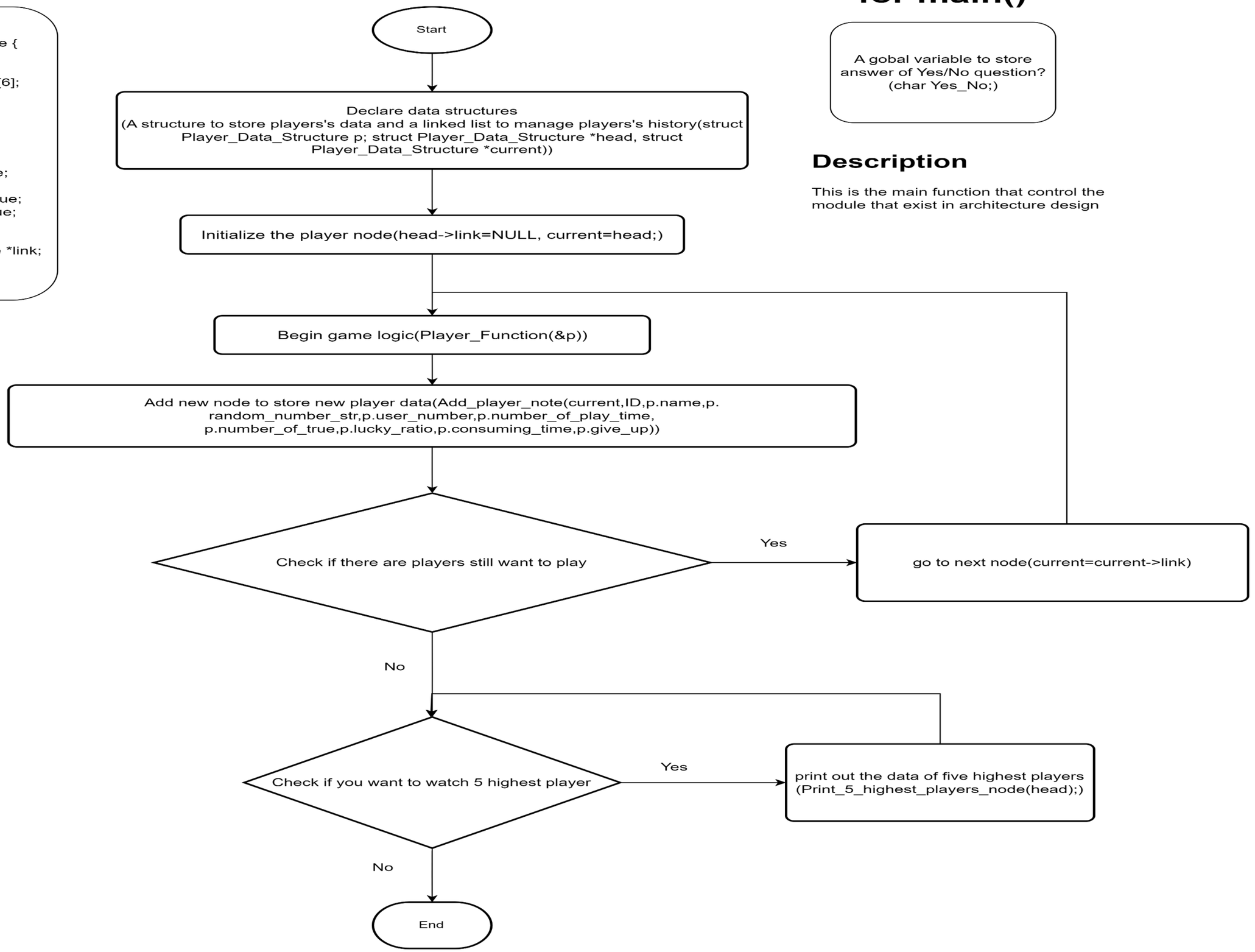
Detail design for main()

```
struct Player_Data_Structure {
    int Player_ID;
    char name[50];
    char random_number_str[6];
    char reserve[3];
    char user_number[6];
    char reserve1[3];
    char result_number[6];
    char reserve2[3];
    char give_up;
    int number_of_play_time;
    int number_of_true;
    int previous_characters_true;
    int current_characters_true;
    float consuming_time;
    float lucky_ratio;
    struct Player_Data_Structure *link;
};
```

A gobal variable to store answer of Yes/No question? (char Yes_No;)

Description

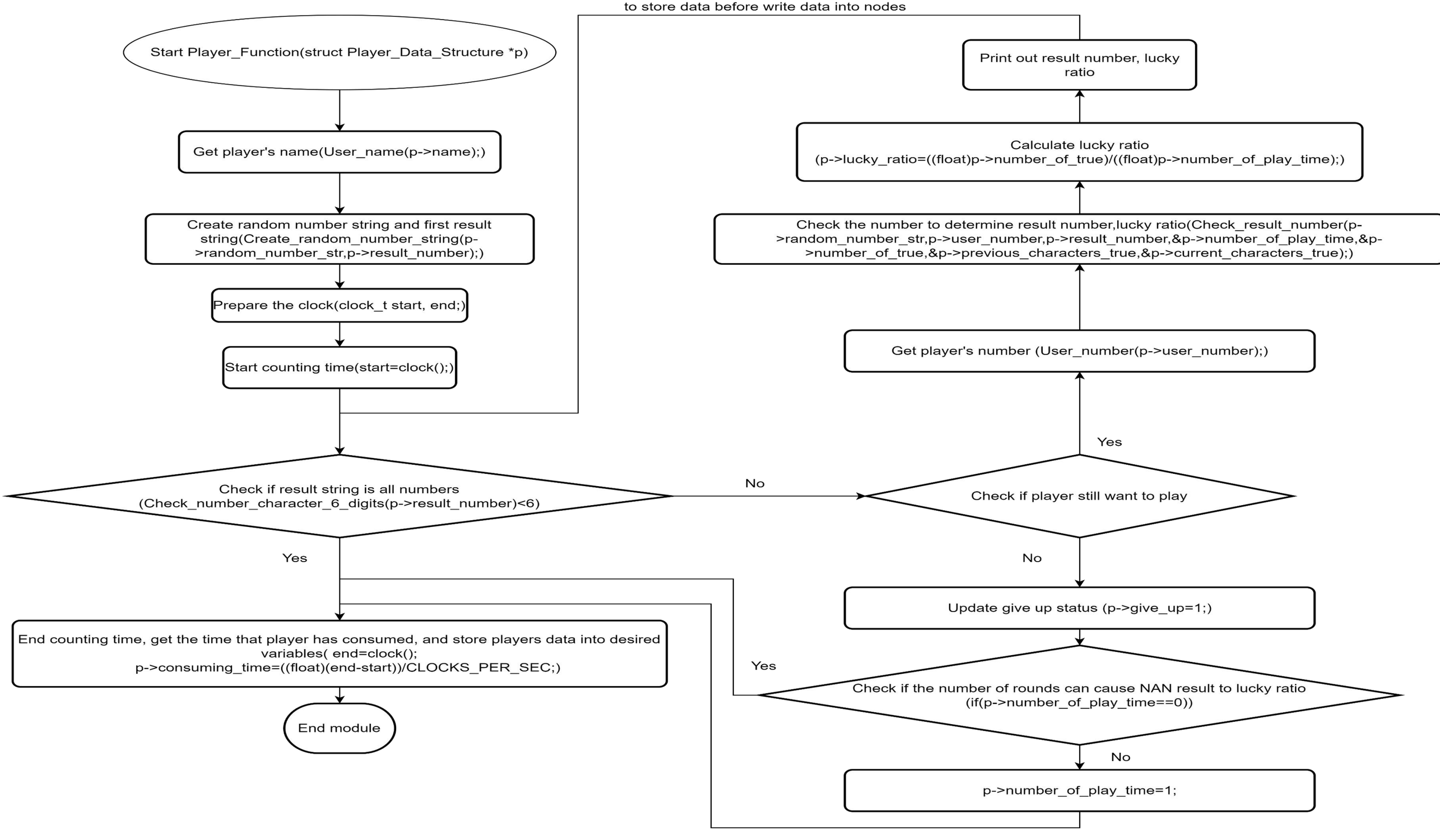
This is the main function that control the module that exist in architecture design



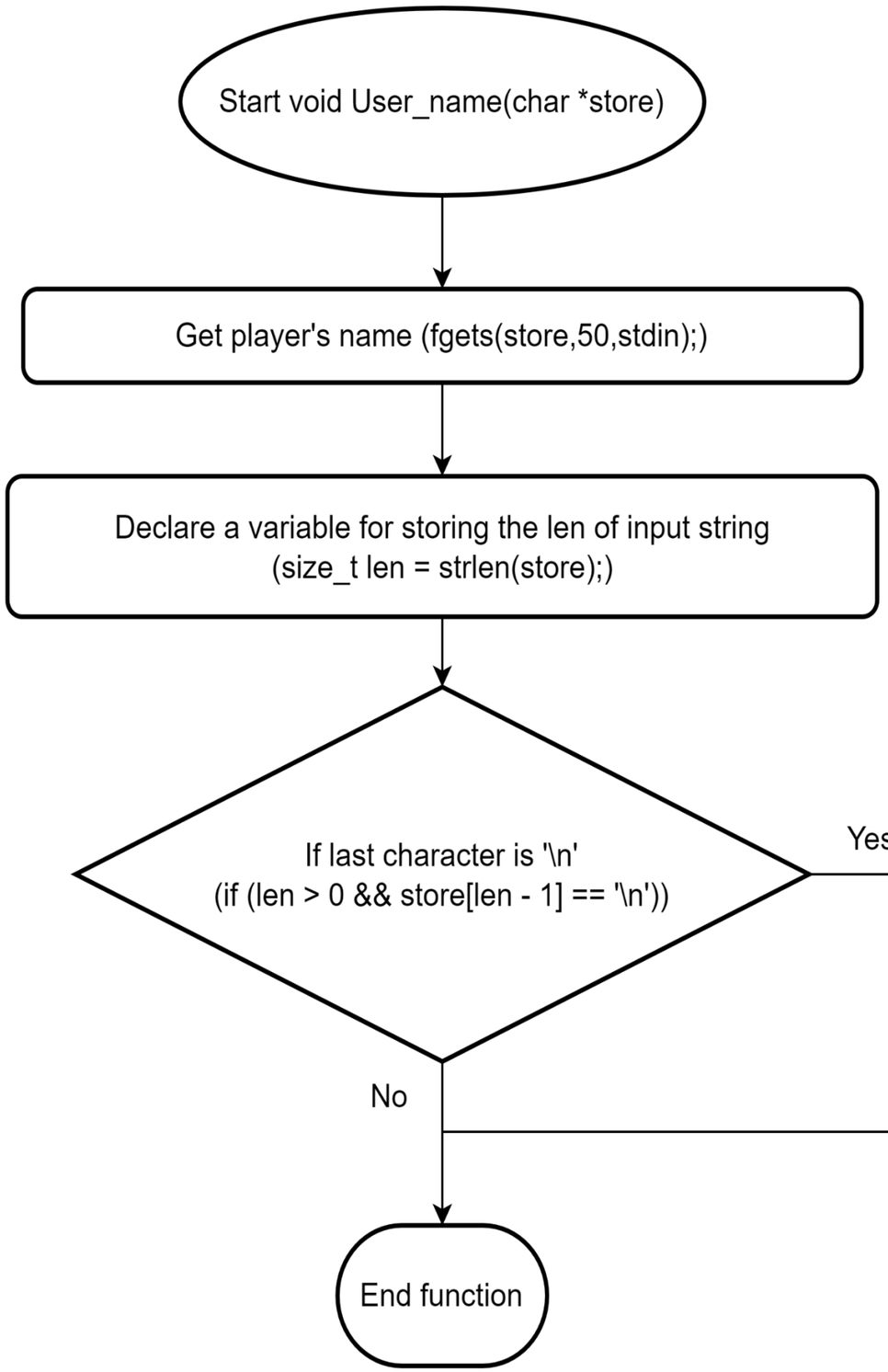
Player_Function() detail design

Description

This is the function that control player logic with the use of timer module.Struct Player_Data_Structure *p is the pointer that points to temporary structure to store data before write data into nodes



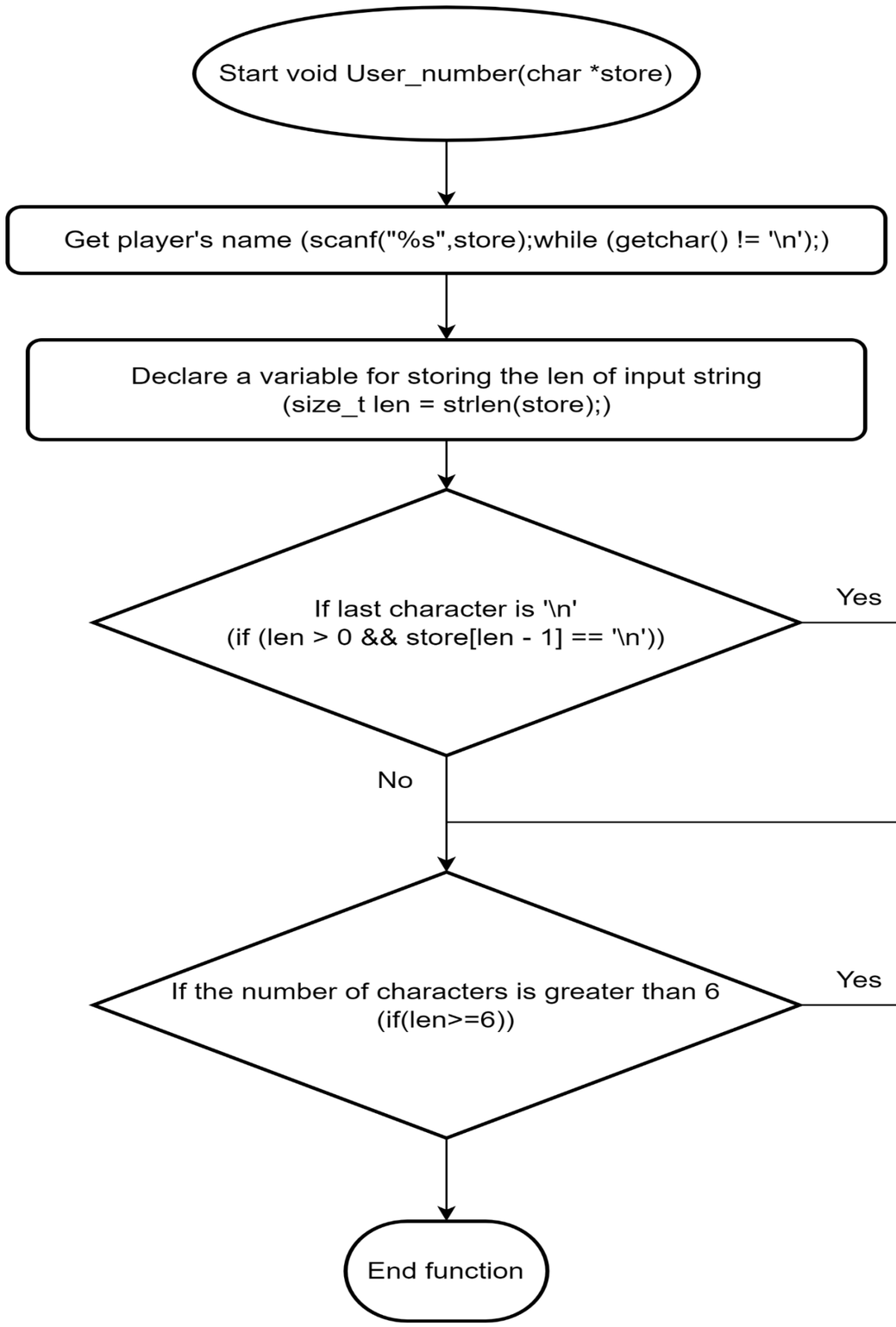
Detail design for User_name



Description

Players enter their names and their name will be stored in a desired location (char *store). We put a pointer to function paramenter to determine the place we want to store player's

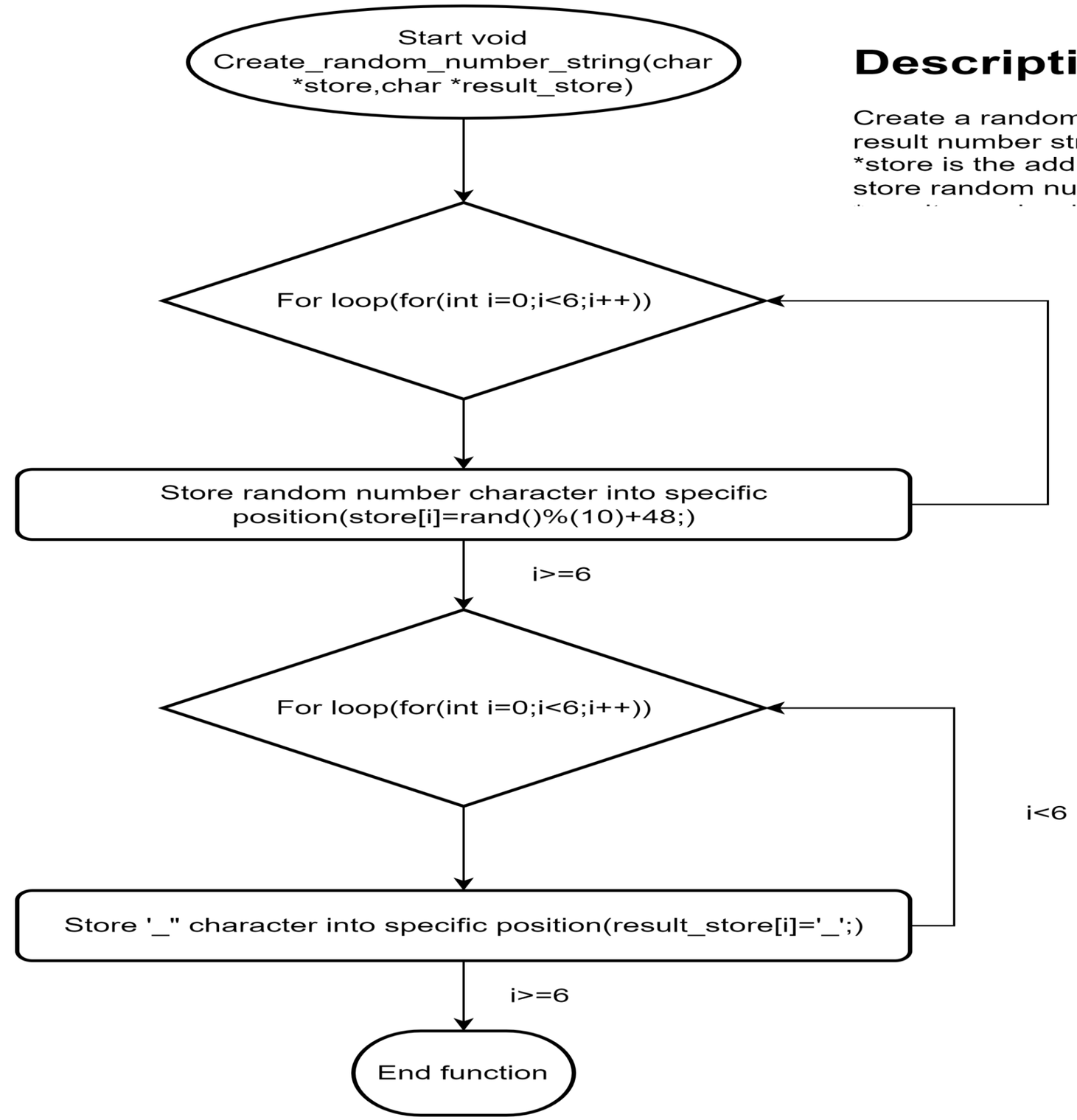
Detail design for User_number



Description

Players enter the number they guest and their answer will be stored in a desired location (char *store). We put a pointer to function paramenter to determine the place we want to

Detail design for Create_random_number_string



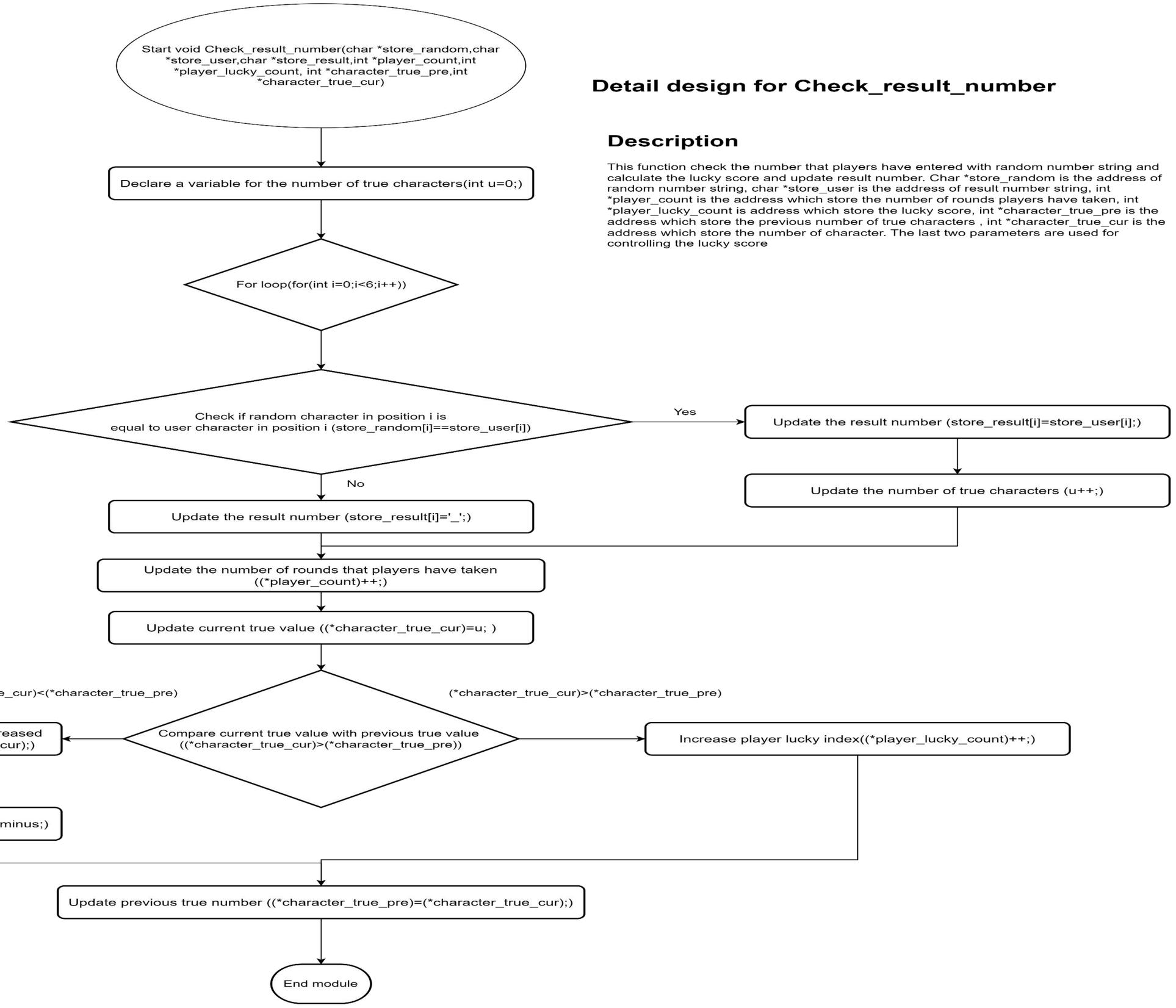
Description

Create a random number in string format and a result number string with value is '_____', char *store is the address of memory that we want to store random number string and char

Detail design for Check_result_number

Description

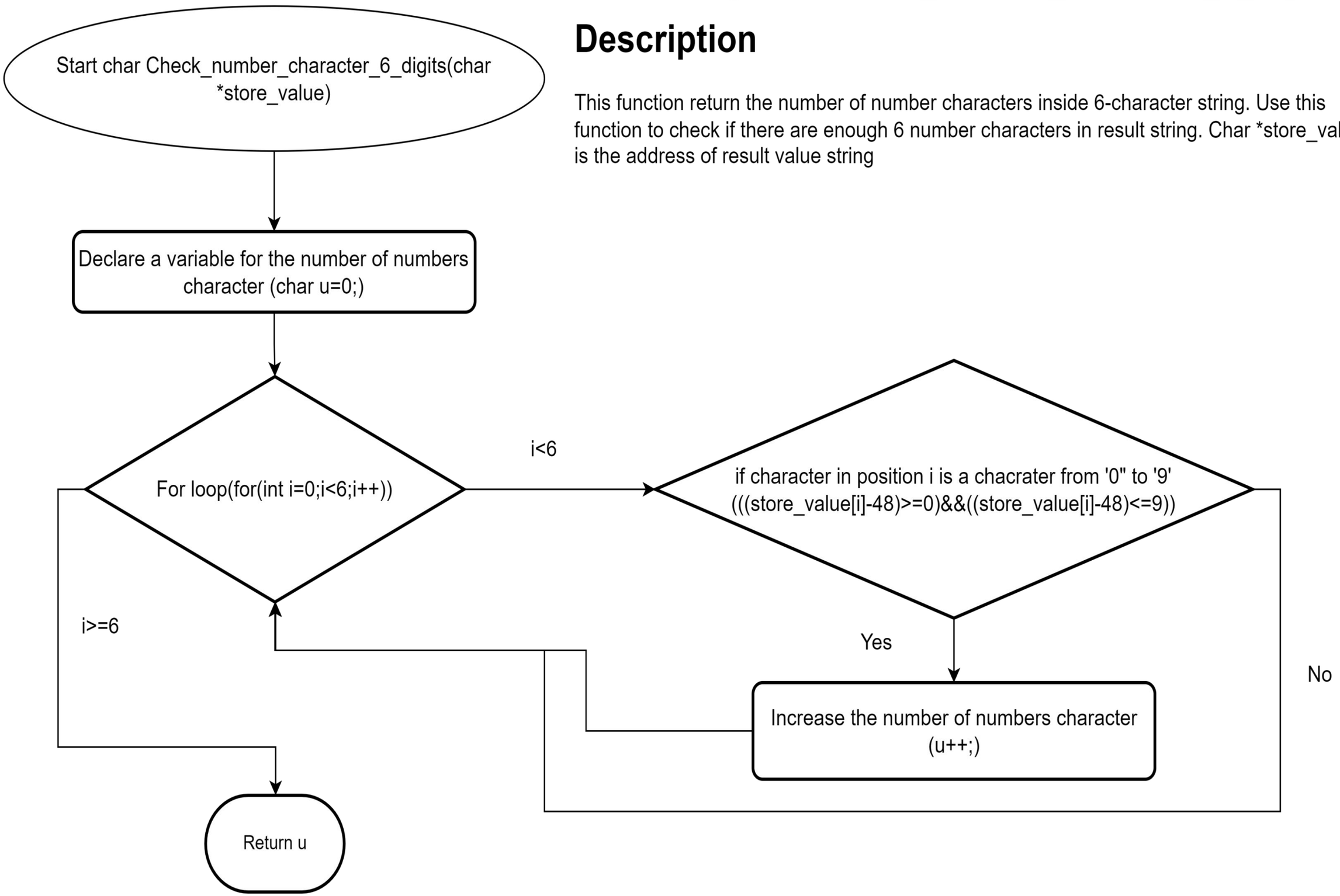
This function check the number that players have entered with random number string and calculate the lucky score and update result number. Char *store_random is the address of random number string, char *store_user is the address of result number string, int *player_count is the address which store the number of rounds players have taken, int *player_lucky_count is address which store the lucky score, int *character_true_pre is the address which store the previous number of true characters , int *character_true_cur is the address which store the number of character. The last two parameters are used for controlling the lucky score



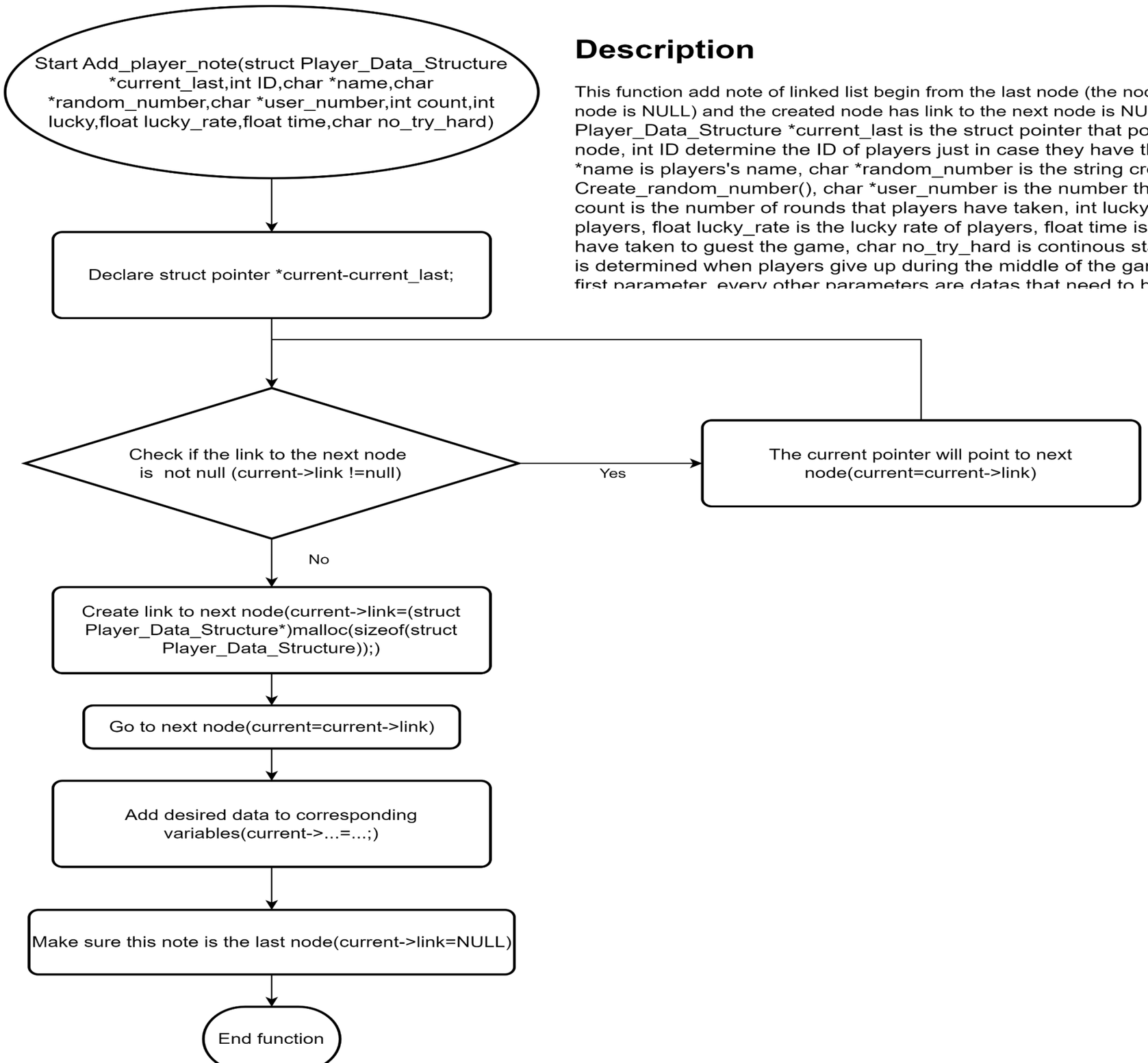
Detail design for Check_number_character_6_digits

Description

This function return the number of number characters inside 6-character string. Use this function to check if there are enough 6 number characters in result string. Char *store_value is the address of result value string



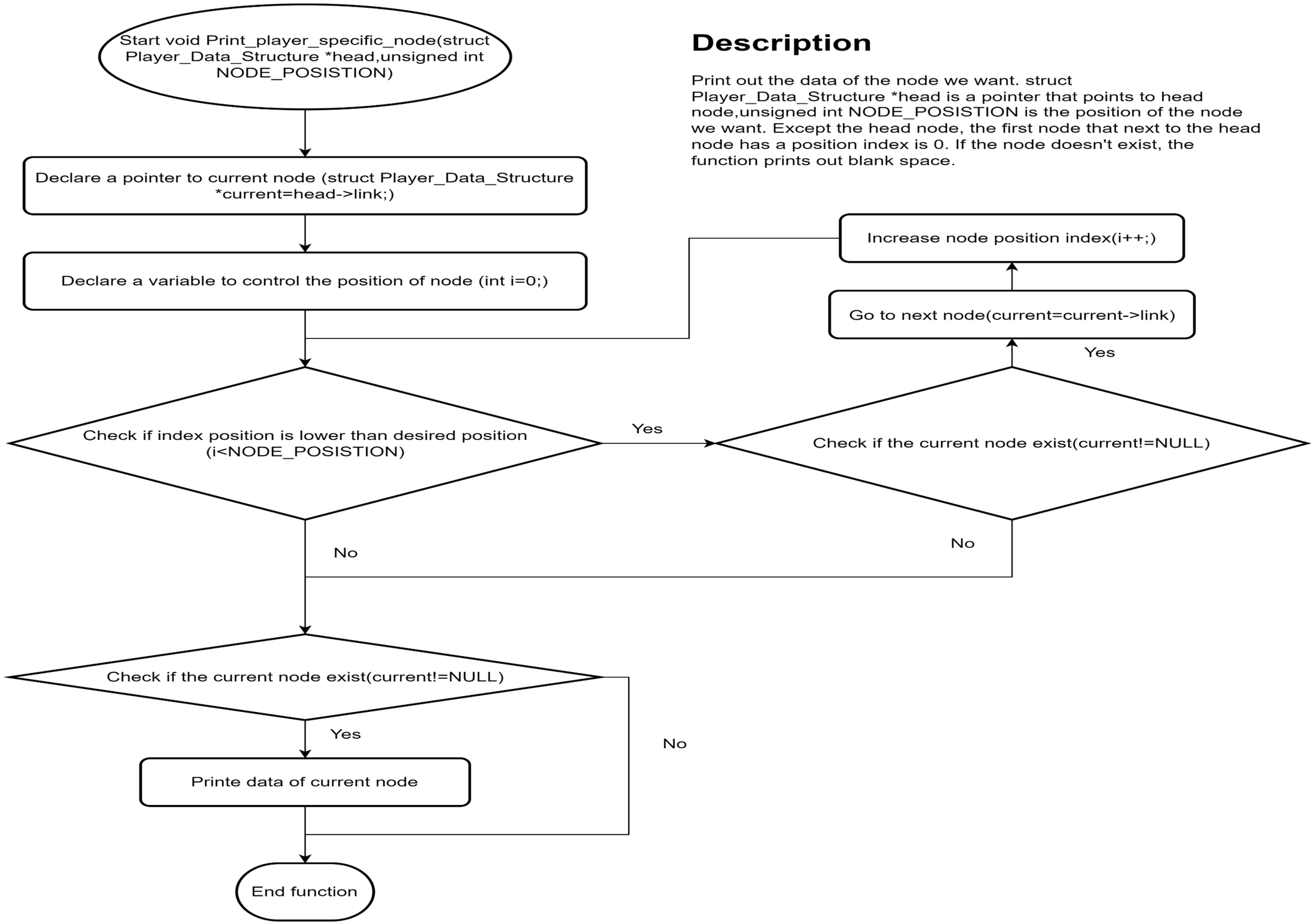
Detail design for Add_player_node



Description

This function add note of linked list begin from the last node (the node with the link to next node is NULL) and the created node has link to the next node is NULL. Struct Player_Data_Structure *current_last is the struct pointer that points to the current node, int ID determine the ID of players just in case they have the same name, char *name is players's name, char *random_number is the string created by Create_random_number(), char *user_number is the number that players enter, int count is the number of rounds that players have taken, int lucky is the lucky score of players, float lucky_rate is the lucky rate of players, float time is the time that players have taken to guest the game, char no_try_hard is continous state of player, this state is determined when players give up during the middle of the game or not. Except the first parameter, every other parameters are datas that need to be written in node

Detail design for Print_player_specific_node



Description

Print out the data of the node we want. struct Player_Data_Structure *head is a pointer that points to head node,unsigned int NODE_POSISTION is the position of the node we want. Except the head node, the first node that next to the head node has a position index is 0. If the node doesn't exist, the function prints out blank space.

Detail design for Print_5_highest_player_node

Description

Print out the data of 5 players with highest ranks base on their lucky ratio,time consumption for player this game. struct Player_Data_Structure *head is a pointer that points to head node. If there are less than 5 players, the lowest ranks will be printed emty.