prototype_list.h

```c
#include<stdbool.h>

#ifndef PROTOTYPE_LIST_H
#define PROTOTYPE_LIST_H

struct PARAMETER{
    char type[10];
    double value;
    struct PARAMETER *next;
    struct PARAMETER *prev;

};

struct PROTOTYPE{
    char funcType[10];
    char funcName[30];
    char libraryName[30];

    struct PARAMETER *paramsHead;
    struct PARAMETER *paramsTail;
    struct PROTOTYPE *prev;
    struct PROTOTYPE *next;
};

// insert import name from full import line
void insertImport(char fulImp[20]);

// returns true if imp is found
bool isImportImported(char *imp);

// prints all included imports
void printAllImports();

// create and returns PARAMETER after creating using type and value
struct PARAMETER* createParameter(const char *type,double value);

// inserts parameter to passed head and tail after creating using type and val
void insertParameter( struct PARAMETER **head, struct PARAMETER **tail, char *type, double val);

// creates proto-type and save it in the list
struct PROTOTYPE* createProto(char *type, char *name, char *libraryName, struct PARAMETER *paramsHead, struct PARAMETER *paramsTail );

// insert proto type to library proto-type list
void insertLibraryProto(struct PROTOTYPE* var);

// prints all library function
void printAllLibraryFunction();

// insert user defined proto-type to list
void insertProto(struct PROTOTYPE* var);

```

```
// returns actual prototype from function call by user, isLibrary true to check library function, false to check user-defined
struct PROTOTYPE* getOriginalProto(struct PROTOTYPE* proto, bool isLibrary);

// checks if passed proto-type exists
bool doesProtoExists(struct PROTOTYPE* proto, bool isLibrary);

// prints all user-defined proto-type
void printAllProto();

// prints proto-type in formatted way, reverse to indicate the parameter order
void printProto(struct PROTOTYPE *ptr, bool reverse);

// returns result after performing library function
double getLibrayFunctionResult(char* name, struct PARAMETER* params);

// for adding library function
void initializeLibraryFunction();

#endif
```