MALAS

Timelimit: 1s

Author: Nathan

Deskripsi Soal : [Author malas membuat deskripsi soal]

Karena author malas, maka kalian diharuskan membuat suatu program yang berfungsi sebagai manajemen dari “TO DO LIST” agar author bisa mengerjakan tugas-tugasnya berdasarkan prioritas tugas tersebut. Program akan memiliki 4 fungsi utama yakni:

1. “ADD” untuk menambahkan tugas baru beserta nilai prioritasnya.
2. “CHECK” untuk menampilkan detail tugas pada indeks input.
3. “LIST” untuk menampilkan semua tugas yang ada.
4. “CLEAR” untuk menyelesaikan dan menghapus tugas pada indeks input.

Input Format :

Serangkaian command beserta parameter inputnya hingga “EXIT”, dengan contoh :

1. **ADD task_name priority**
2. **CHECK [id]**
3. **LIST**
4. **CLEAR [id]**
5. **EXIT**

Output Format :

Respons berupa informasi program setelah menjalankan command yang berkaitan, contohnya :

1. Untuk **“ADD task_name priority”** : **“task_name priority is added”**
2. Untuk **“CHECK [id]” :**
   Jika tugas ada / tidak dihapus : **“task_name priority”**
   Jika tugas sudah dihapus : **“Task at ID [id] is empty”**

3. Untuk **“LIST”** :
   “
   TO DO LIST:
   task_name1 priority1
   task_name2 priority2
   …
   task_name priority
   ”
4. Untuk **“CLEAR [id]”** :
   “Task task_name is cleared”

Constraint :

Number of Command < 50

0 < strlen(task_name) <= 50; dipastikan tidak ada spasi dalam nama

-100 < priority < 100

Input Example :

```
ADD kasih_makan_kucing 100
ADD praktikum_sbd 90
ADD praktikum_dasprog -10
ADD panit_schematics -30
LIST
CLEAR 1
CHECK 0
CHECK 1
LIST
EXIT
```

Output Example :

```
kasih_makan_kucing 100 is added
praktikum_sbd 90 is added
praktikum_dasprog -10 is added
panit_schematics -30 is added
TO DO LIST:
kasih_makan_kucing 100
praktikum_sbd 90
praktikum_dasprog -10
panit_schematics -30
Task praktikum_sbd is cleared
kasih_makan_kucing 100
Task at ID 1 is empty
TO DO LIST:
kasih_makan_kucing 100
praktikum_dasprog -10
panit_schematics -30
```

NOTE : WAJIB GUNAKAN TEMPLATE DI BAWAH INI !!!

NOTE : MUST BE USE THE TEMPLATE BELOW !!!

Implementasi wajib menggunakan Dynamic Memory Allocation melalui realloc.

Implementation must using the Dynamic Memory Allocation using the realloc.

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

typedef struct {
   char name[50];
   int priority;
} task;

int main(){
   task *taskPtr = NULL;
   int index = 0;
   while (1){
      char cmd[20];
      scanf(" %s", cmd);
      if (!strcmp(cmd, "EXIT")) {
         break;
      }
      else if (!strcmp(cmd, "ADD")){
         // realloc *taskPtr
         scanf(" %s %d", taskPtr[index].name, &taskPtr[index].priority);
         index++;
      }
      else if (!strcmp(cmd, "CHECK")){
         //...
      }
      else if (!strcmp(cmd, "LIST")){
         //...
      }
      else if (!strcmp(cmd, "CLEAR")){
         //...
      }
```

MALAS

Timelimit: 1s

Author: Nathan

Problem Description : [Author too lazy to write the problem description]

Because the author is lazy, you need to make a program that functioned as a management of "TO DO LIST" system so that the author could doing all his tasks based on the task priority. Program will have 4 main functions :

1. "ADD" to add new task with it's priority value.
2. "CHECK" to display the task detail based on the index input.
3. "LIST" to display all tasks.
4. "CLEAR" to clear and delete the task at the index input.

Input Format :

Order of command with it's input parameter until "EXIT", for example :

1. **ADD task_name priority**
2. **CHECK [id]**
3. **LIST**
4. **CLEAR [id]**
5. **EXIT**

Output Format :

Respons about the program information when executing the relevant command, for example :

1. For **"ADD task_name priority"** : **"task_name priority is added"**
2. For **"CHECK [id]" :**
   If the task is exist / not cleared : **"task_name priority"**
   If the task is cleared / deleted : **"Task at ID [id] is empty"**
3. For **"LIST"** :

“

TO DO LIST:
task_name1 priority1
task_name2 priority2
…
task_name priority
”

4. For **“CLEAR [id]”** :
   “Task task_name is cleared”

Constraint :

Number of Command < 50

0 < strlen(task_name) <= 50; there is no space between the character in the task name

-100 < priority < 100

Input Example :

```
ADD kasih_makan_kucing 100
ADD praktikum_sbd 90
ADD praktikum_dasprog -10
ADD panit_schematics -30
LIST
CLEAR 1
CHECK 0
CHECK 1
LIST
EXIT
```

Output Example :

```
kasih_makan_kucing 100 is added
praktikum_sbd 90 is added
praktikum_dasprog -10 is added
panit_schematics -30 is added
TO DO LIST:
kasih_makan_kucing 100
praktikum_sbd 90
praktikum_dasprog -10
panit_schematics -30
Task praktikum_sbd is cleared
kasih_makan_kucing 100
Task at ID 1 is empty
TO DO LIST:
kasih_makan_kucing 100
praktikum_dasprog -10
panit_schematics -30
```