

ASSIGNMENT1: MEMORY MANAGEMENT

DUE DATE: 11:59 PM , 3rd February (Monday night)

extern char *p; It is a pointer pointing to the char array

extern int size; Size of char array

Functions to be implemented:

Note: You are not allowed to change the function definitions in the .h file. We have provided to you the client and the .h file ,your job is to write and submit only the implementation file.

void allocate(int n);

This function allocates memory of n bytes for the global character array. The mymalloc function must allocate memory only from this array. The memory for bookkeeping as well must also be allocated within those n bytes itself.

void* mymalloc(int SIZE);

This should return the pointer that refers to the free memory location of **SIZE** bytes. Note this pointer must be chosen based on what policy you are implementing (i.e.: first fit, best fit or worst fit).
If no free memory block is found return NULL.

void myfree(void *b);

Free the block pointed by the parameter.
It should be able to merge two free blocks.
If the pointer does not point to any block it should do nothing.

void print_book();

Print size used by book-keeping structure.

void display_mem_map();

// Assume n = 1000

// assume bookkeeping : 10 bytes for both allocated and free cases

// Print a tab space (\t) between the columns

It should print the following if called at start

start_addr	block_in_bytes	status
0	10	book
10	990	free

Sample output with the sanity_client:

```

karan@DESKTOP-LH4HECJ: /mnt/d/pes stuff/sem 4/fine techniques/fine_techniques/assignment_1$ gcc 1142_A1.c sanity_client.c
karan@DESKTOP-LH4HECJ: /mnt/d/pes stuff/sem 4/fine techniques/fine_techniques/assignment_1$ ./sanity_client.c
0      16      book
16     30      allocated
46     16      book
52     138     free
0      16      book
16     30      allocated
46     16      book
62     40      allocated
102    16      book
118    82      free
0      16      book
16     30      allocated
46     16      book
62     40      allocated
102    16      book
118    30      allocated
148    16      book
164    36      free
0      16      book
16     30      free
46     16      book
62     40      allocated
102    16      book
118    30      allocated
148    16      book
164    36      free
0      16      book
16     86      free
102    16      book
118    30      allocated
148    16      book
164    36      free
0      16      book
16     184     free

```

In case there's no space to keep a bookkeeping structure for a small amount of free space, make sure that the free space is allocated to the block before (allocated block on the left) , and the same should be displayed by display_mem_map. The following screenshot should explain what we mean by that:

```
1 #include <stdio.h>
2 #include "assignment_1.h"
3 void main()
4 {
5     allocate(78);
6     char *a = (char *)malloc(10 * sizeof(char));
7     char *b = (char *)malloc(10 * sizeof(char));
8     char *c = (char *)malloc(10 * sizeof(char));
9     display_mem_map();
10    printf("\n");
11
12    myfree(b);
13    display_mem_map();
14    printf("\n");
15
16    b = (char *)malloc(5 * sizeof(char));
17    display_mem_map();
18    printf("\n");
19 }
20
21
```

In the above example, the pointer b has got those 5 extra bytes where we couldn't fit a new book.

Each student will implement one allocation policy. Following are the list of allocation policies:

1. first fit (FF)
2. best fit (BF)
3. worst fit (WF)

$((\text{last four digits of SRN}) \% 3) + 1$ will be the allocation policy you will be implementing.

Deliverables: Two files (.c) and (.txt)

A1_<Allocation Policy>_<SRN>.c: This should contain the implementation for the above functions. Please ensure that if you are using any helper functions in your implementation file, they are named with the static keyword so they don't clash with any functions that we will use to test your code.

Please comment your code as well (you will be graded based on this as well).

Please use the sanity_client.c to check if your implementation works.

E.g.: A1_FF_PES1201701142.c or A1_BF_PES1201701142.c or

A1_WF_PES1201701142.c for first fit, best fit and worst fit respectively.

A1_README_<SRN>.txt: This should contain the following things (keep these short and precise, we need to go through ~100 of these 😊):

- The allocation policy implemented by you.
- Basic logic used while implementing the policy.
- Explanation of the book keeping structure.
- Key takeaway from this assignment.
- Any additional notes.

You can ask doubts at: [Doubts clarifications](#)

Submission Link: <https://forms.gle/VCTtbKcsDbWWdiUt7>

Any cases of plagiarism will be dealt with seriously. DO NOT COPY/SHARE CODE. Do not take code from your peers or seniors.