C_resit_2021_partA

Computation I: hardware/software interface (Technische Universiteit Eindhoven)

Computation I 5EIA1
C Exam Part 1: Registry (v1.2, January 15, 2021)
19 January 18:00-19:40

In this exam you will develop a registry program. You can add and remove persons.

Important

- In this exam a predefined function is available for each task, e.g. `predefined_insert`. Therefore, if you get stuck on a task or want to skip it then you can use the predefined function instead of your own function.
- If you use the predefined function for a task anywhere in your code then you will not get points for all of the test cases of that task. For example, if instead of writing your own `insert` in Task 1 you use `predefined_insert` in later tasks then you will not get the points for test cases 2 and 4 to 6 for Task 1. You will get points for the other test cases that use `predefined_insert`.
- Oncourse will show all the test cases passed, including those using the predefined functions. Points for using predefined functions will be deducted after the exam.
- To use the predefined functions you need to include `#include "predefined.h"` in your program. This is done automatically when you create a new .c file.

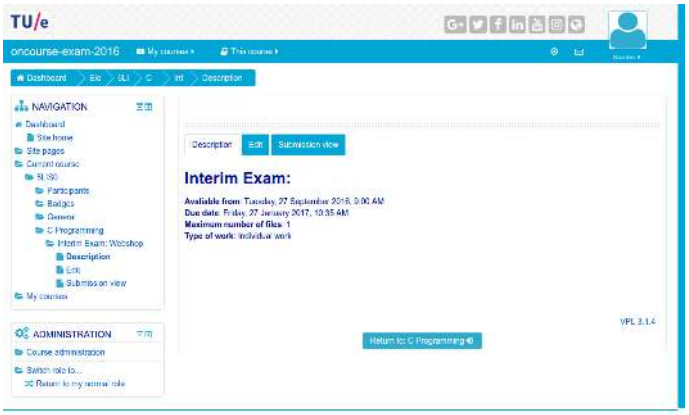
function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	% per fn	cumulative %
quit	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7%	7%
insert		1		1	1	1	1	1	1	1	1	1	1	1	29%	36%
print			1	1	1	1	1	1	1	1	1	1	1	1	7%	43%
correct case							1	1							14%	57%
delete									1	1	1	1			29%	86%
print reverse													1	1	14%	100%

Figure 1: Test cases and points per task. You can use predefined functions to not implement a function yourself but you will not get the points for that function. All test cases are listed at the end of this document.

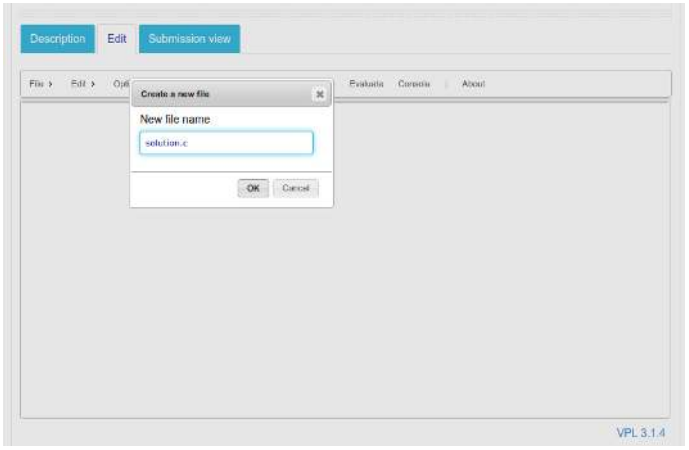
Important

- Your grade is based on the number of cases passed. Try to complete the cases one by one.
- You can press "evaluate" as often as you like during the exam to evaluate your solution. We advise you to do this regularly during the exam.
- You must follow the instructions of the exam. For example, you may not use an array if a linked-list implementation is asked for.
- You are only allowed to use materials offered electronically as part of the exam (cheat sheet and K&R book) during the exam. No other electronic or printed material are allowed.
- The grade is based on your **last submission**. Make sure you submit a working version that completes as many cases as possible. It's also wise not to make last-minute changes.
- Your program must work for all inputs, not just the test cases. For example, when a `#define MAX 10` must be defined as part of the exam, then your program should work for all values of `MAX`. We will change the test cases after the exam (but the number of test cases per task, i.e., the grade of correct programs will not change).

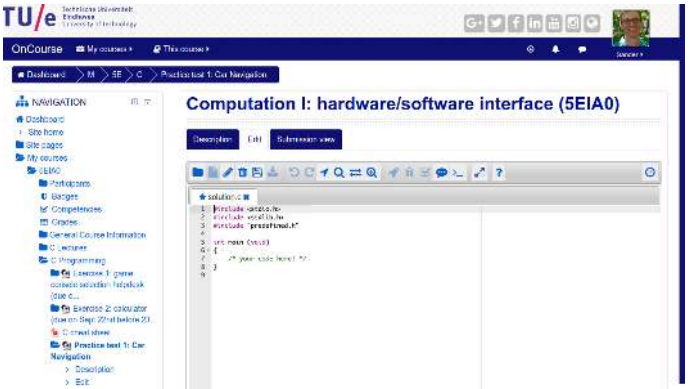
Open the correct C programming exam on exam.oncourse.tue.nl. You will see a screen similar to:



Select the "Edit" tab and the following screen will appear:



Provide a useful name for your C file (e.g., solution.c) and press "Ok". **The file name must end in .c and cannot contain any spaces.** You will then see the following screen:



You can write your own C program in the text editor that is now shown in your browser. Once you press "Save" you can "Run" and "Evaluate" your program. Using the "Run" command you will see a terminal where you can provide input to your program. You can use this to debug your program. Using the "Evaluate" command all test cases are evaluated and at the end the results are displayed (grade and errors if present).

Task 1. Write a C program that asks the user to select the command that needs to be performed. The commands that need to be supported are listed in the following table:

Character	Command
q	quit
i	insert person in a sorted list
p	print the list of all persons
c	correct the case of names
d	delete all persons below a given height
r	print the list of all persons in reverse order

In this task you only need to implement the quit command. If an invalid character is given then print the error message `Unknown command 'x'` (with x replaced by the unknown command):

```
Command? z
Unknown command 'z'.
Command? q
Bye!
```

Your program must produce the *exact* output, including all spaces, capitalisation, quotes, etc.

Your program should now pass test case 1.

The `predefined.h` file that you should include in your program with `#include "predefined.h"` contains the following declarations:

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

struct person {
  char *name;
  float height;
  struct person *next;
};

extern struct person *predefined_insert (struct person *registry, char *name, float height);
extern void predefined_print_registry (struct person *registry);
```

Task 2. In your main function define a variable `registry` of type pointer to `struct person` to keep track of all persons in a linked list. Add the 'i' command to insert a new person in the registry. Write the function `struct person *insert(struct person *registry, char *name, float height)` that adds a new person with the given name and height in the list. The list is *ordered first by height and then by name*, as shown in the example output below. Names do not contain spaces; in other words, you can use the `"%s"` format string for `scanf`.

```
Command? i
name? petitsuix
height? 10.0
Command? i
name? numeronobis
height? 9.00
Command? i
name? justforkix
height? 11
Command? i
name? dairiprodus
height? 19.5
Command? i
name? vitriolix
height? 19.5
Command? i
name? giveusabonus
height? 19.5
... print statement to be implemented in the next task ...
Command? p
numeronobis is 9.00 cm tall
petitsuix is 10.00 cm tall
justforkix is 11.00 cm tall
dairiprodus is 19.50 cm tall
giveusabonus is 19.50 cm tall
vitriolix is 19.50 cm tall
Command? q
Bye!
```

Your program should now pass test cases 1 to 2. If you do not wish to implement the function `insert` then you can use the `predefined_insert` function, but you will not get the points for test cases 2, 4, 5, 6.

You can call the predefined function just like your own function:
`predefined_insert(registry, name, height);`

Hint: If when you run your program you receive an error message similar to:

```
/tmp/ccJmRCOb.o: In function ‘predefined_insert’:
  exam.c:(.text+0x0): multiple definition of ‘predefined_insert’
```

then you have named your own function `predefined_insert` instead of `insert`. When you write your own function then it should not contain `predefined_` in the name.

Task 3. Add the 'p' command to print all persons with your `void print_registry(struct person *registry)` function. The height is printed with two digits after the decimal point.

```
Command? p
Command? i
name? zaphod
height? 170
Command? i
name? rontok
height? 203
Command? p
zaphod is 170.00 cm tall
rontok is 203.00 cm tall
Command? q
Bye!
```

Hint: The digits after the decimal point are specified by the *precision* in the format string, described in Kernighan & Ritchie Appendix B1.2.

Your program should now pass test cases 1 to 6. If you do not wish to implement the function `print_registry` then you can use the `predefined_print_registry` function, but you will not get the points for test case 3.

Note that the following tasks are independent and you can make them in any order you wish.

Task 4. Add the 'c' command to correct the capitalisation of the names of the persons in the registry. Every person's name should start with a capital letter ('A' to 'Z') and be followed by lower-case letters ('a' to 'z'). Write the function `void correct_case(struct person *registry)` to accomplish this.

```
Command? i
name? nebulus
height? 123
Command? i
name? Confoundtheirpolitix
height? 234
Command? i
name? pSYCHOANALYTIX
height? 123.4
Command? p
nebulus is 123.00 cm tall
pSYCHOANALYTIX is 123.40 cm tall
Confoundtheirpolitix is 234.00 cm tall
Command? c
Command? p
Nebulus is 123.00 cm tall
Psychoanalytix is 123.40 cm tall
Confoundtheirpolitix is 234.00 cm tall
Command? q
Bye!
```

Hint: Although not required, the string functions in Kernighan & Ritchie Appendix B2 may be useful.

Your program should now pass test cases 1 to 8. Since no other function depends on `correct_case` there is no predefined version for it.

Task 5. Add the 'd' command to delete all persons that are strictly smaller than the given minimum height from the registry, using your `struct person *delete_height (struct person *registry, float height)` function. You must free any space that has been previously malloc'd.

```
Command? i
name? tall
height? 210
Command? i
name? short
height? 150
Command? i
name? average
height? 178
Command? p
short is 150.00 cm tall
average is 178.00 cm tall
tall is 210.00 cm tall
Command? d
minimum height? 150
Command? p
short is 150.00 cm tall
average is 178.00 cm tall
tall is 210.00 cm tall
Command? d
minimum height? 180
Command? p
tall is 210.00 cm tall
Command? q
Bye!
```

Hint: You may want to first find the first person that is tall enough, and then delete all persons until then.

Your program should now pass test cases 1 to 12. Since no other function depends on `delete_height` there is no predefined version for it.

Task 6. Add the 'r' command to print all persons with the `void print_registry_reverse(struct person *registry)` function, but *in reverse order*.

```
Command? i
name? dairiprodus
height? 9
Command? i
name? vitrolix
height? 8
Command? i
name? nautilus
height? 13
Command? p
vitrolix is 8.00 cm tall
dairiprodus is 9.00 cm tall
nautilus is 13.00 cm tall
Command? r
nautilus is 13.00 cm tall
dairiprodus is 9.00 cm tall
vitrolix is 8.00 cm tall
Command? q
Bye!
```

Hint: Recursion.

Your program should now pass test cases 1 to 14. Since no other function depends on `print_registry_reverse` there is no predefined version for it.

Submission: Your *last* submission will be graded. Note that points will be deducted after the exam for using predefined functions.

Input / output test cases

Long lines have been wrapped at 70 characters for legibility. When your program output is compared to the expected output lines will not be wrapped.

Case 01

Input:

```
q
```

Output:

```
Command? Bye!
```

Case 02

Input:

```
i
asterix
170
i
getafix
170.23
i
vitalstatistix
203
q
```

Output:

```
Command? name? height? Command? name? height? Command? name? height?
Command? Bye!
```

Case 03

Input:

```
p
p
q
```

Output:

```
Command? Command? Command? Bye!
```

Case 04

Input:

```
p
i
zaphod
170
p
i
humma
170.23
p
i
rontok
203
p
q
```

Output:

```
Command? Command? name? height? Command? zaphod is 170.00 cm tall
Command? name? height? Command? zaphod is 170.00 cm tall
humma is 170.23 cm tall
Command? name? height? Command? zaphod is 170.00 cm tall
humma is 170.23 cm tall
rontok is 203.00 cm tall
Command? Bye!
```

Case 05

Input:

```
i
asterix
170
i
getafix
170.23
i
cacofonix
203
p
i
asterix
200
p
i
geriatrix
203
p
i
vitalstatistix
203
p
q
```

Output:

```
Command? name? height? Command? name? height? Command? name? height?
Command? asterix is 170.00 cm tall
getafix is 170.23 cm tall
cacofonix is 203.00 cm tall
Command? name? height? Command? asterix is 170.00 cm tall
getafix is 170.23 cm tall
asterix is 200.00 cm tall
cacofonix is 203.00 cm tall
Command? name? height? Command? asterix is 170.00 cm tall
getafix is 170.23 cm tall
asterix is 200.00 cm tall
cacofonix is 203.00 cm tall
geriatrix is 203.00 cm tall
Command? name? height? Command? asterix is 170.00 cm tall
getafix is 170.23 cm tall
asterix is 200.00 cm tall
cacofonix is 203.00 cm tall
geriatrix is 203.00 cm tall
vitalstatistix is 203.00 cm tall
Command? Bye!
```

Case 06

Input:

```
i
petitsuix
10
i
numeronobis
9
i
justforkix
11
p
i
dairiprodus
19.5
i
vitriolix
19.5
i
giveusabonus
19.5
p
i
nautilus
3.5
i
crustacius
3.5
i
nefarius
3.5
p
q
```

Output:

```
Command? name? height? Command? name? height? Command? name? height?
Command? numeronobis is 9.00 cm tall
petitsuix is 10.00 cm tall
justforkix is 11.00 cm tall
Command? name? height? Command? name? height? Command? name? height?
Command? numeronobis is 9.00 cm tall
petitsuix is 10.00 cm tall
justforkix is 11.00 cm tall
dairiprodus is 19.50 cm tall
giveusabonus is 19.50 cm tall
vitriolix is 19.50 cm tall
Command? name? height? Command? name? height? Command? name? height?
Command? crustacius is 3.50 cm tall
nautilus is 3.50 cm tall
nefarius is 3.50 cm tall
numeronobis is 9.00 cm tall
petitsuix is 10.00 cm tall
justforkix is 11.00 cm tall
dairiprodus is 19.50 cm tall
giveusabonus is 19.50 cm tall
vitriolix is 19.50 cm tall
Command? Bye!
```

Case 07

Input:

```
i
nebulus
123
i
Confoundtheirpolitix
234
i
pSYCHOANALYTIX
123.4
p
c
p
q
```

Output:

```
Command? name? height? Command? name? height? Command? name? height?
Command? nebulus is 123.00 cm tall
pSYCHOANALYTIX is 123.40 cm tall
Confoundtheirpolitix is 234.00 cm tall
Command? Command? Nebulus is 123.00 cm tall
Psychoanalytix is 123.40 cm tall
Confoundtheirpolitix is 234.00 cm tall
Command? Bye!
```

Case 08

Input:

```
i
b
100
i
A
100
i
AhAhAhAhAh
100
i
bB
100
p
c
p
q
```

Output:

```
Command? name? height? Command? name? height? Command? name? height?
Command? name? height? Command? A is 100.00 cm tall
AhAhAhAhAh is 100.00 cm tall
b is 100.00 cm tall
bB is 100.00 cm tall
Command? Command? A is 100.00 cm tall
Ahahahahah is 100.00 cm tall
B is 100.00 cm tall
Bb is 100.00 cm tall
Command? Bye!
```

Case 09

Input:

```
p
d
100
p
i
noone
90
p
d
100
p
q
```

Output:

```
Command? Command? minimum height? Command? Command? name? height?
Command? noone is 90.00 cm tall
Command? minimum height? Command? Command? Bye!
```

Case 10

Input:

```
i
petitsuix
10
i
numeronobis
11
i
justforkix
12
p
i
dairiprodus
9
i
vitriolix
8
i
giveusabonus
7
i
nautilus
13
p
d
1
p
q
```

Output:

```
Command? name? height? Command? name? height? Command? name? height?
Command? petitsuix is 10.00 cm tall
numeronobis is 11.00 cm tall
justforkix is 12.00 cm tall
Command? name? height? Command? name? height? Command? name? height?
Command? name? height? Command? giveusabonus is 7.00 cm tall
vitriolix is 8.00 cm tall
dairiprodus is 9.00 cm tall
petitsuix is 10.00 cm tall
numeronobis is 11.00 cm tall
justforkix is 12.00 cm tall
nautilus is 13.00 cm tall
Command? minimum height? Command? giveusabonus is 7.00 cm tall
vitriolix is 8.00 cm tall
dairiprodus is 9.00 cm tall
petitsuix is 10.00 cm tall
numeronobis is 11.00 cm tall
justforkix is 12.00 cm tall
nautilus is 13.00 cm tall
Command? Bye!
```

Case 11

Input:

```
i
petitsuix
10
i
numeronobis
11
i
justforkix
12
p
i
dairiprodus
9
i
vitriolix
8
i
giveusabonus
7
i
nautilus
13
p
d
7
p
d
8
p
q
```

Output:

```
Command? name? height? Command? name? height? Command? name? height?
Command? petitsuix is 10.00 cm tall
numeronobis is 11.00 cm tall
justforkix is 12.00 cm tall
Command? name? height? Command? name? height? Command? name? height?
Command? name? height? Command? giveusabonus is 7.00 cm tall
vitriolix is 8.00 cm tall
dairiprodus is 9.00 cm tall
petitsuix is 10.00 cm tall
numeronobis is 11.00 cm tall
justforkix is 12.00 cm tall
nautilus is 13.00 cm tall
Command? minimum height? Command? giveusabonus is 7.00 cm tall
vitriolix is 8.00 cm tall
dairiprodus is 9.00 cm tall
petitsuix is 10.00 cm tall
numeronobis is 11.00 cm tall
justforkix is 12.00 cm tall
nautilus is 13.00 cm tall
Command? minimum height? Command? vitriolix is 8.00 cm tall
dairiprodus is 9.00 cm tall
petitsuix is 10.00 cm tall
numeronobis is 11.00 cm tall
justforkix is 12.00 cm tall
nautilus is 13.00 cm tall
Command? Bye!
```

Case 12

Input:

```
i
petitsuix
10
i
numeronobis
11
i
justforkix
12
p
i
dairiprodus
9
i
vitriolix
8
i
giveusabonus
7
i
nautilus
13
p
d
13
p
i
vitalstatistix
13
i
abraracourix
13
i
getafix
13
p
d
13
p
d
14
p
q
```

Output:

```
Command? name? height? Command? name? height? Command? name? height?
Command? petitsuix is 10.00 cm tall
numeronobis is 11.00 cm tall
justforkix is 12.00 cm tall
Command? name? height? Command? name? height? Command? name? height?
Command? name? height? Command? giveusabonus is 7.00 cm tall
vitriolix is 8.00 cm tall
dairiprodus is 9.00 cm tall
petitsuix is 10.00 cm tall
numeronobis is 11.00 cm tall
justforkix is 12.00 cm tall
nautilus is 13.00 cm tall
Command? minimum height? Command? nautilus is 13.00 cm tall
Command? name? height? Command? name? height? Command? name? height?
Command? abraracourix is 13.00 cm tall
getafix is 13.00 cm tall
nautilus is 13.00 cm tall
vitalstatistix is 13.00 cm tall
Command? minimum height? Command? abraracourix is 13.00 cm tall
getafix is 13.00 cm tall
nautilus is 13.00 cm tall
vitalstatistix is 13.00 cm tall
Command? minimum height? Command? Command? Bye!
```

Case 13

Input:

```
r
i
petitsuix
10
r
i
numeronobis
11
r
i
justforkix
12
i
dairiprodus
9
i
vitriolix
8
i
giveusabonus
7
i
nautilus
13
r
q
```

Output:

```
Command? Command? name? height? Command? petitsuix is 10.00 cm tall
Command? name? height? Command? numeronobis is 11.00 cm tall
petitsuix is 10.00 cm tall
Command? name? height? Command? name? height? Command? name? height?
Command? name? height? Command? name? height? Command? nautilus is
13.00 cm tall
justforkix is 12.00 cm tall
numeronobis is 11.00 cm tall
petitsuix is 10.00 cm tall
dairiprodus is 9.00 cm tall
vitriolix is 8.00 cm tall
giveusabonus is 7.00 cm tall
Command? Bye!
```

Case 14

Input:

```
i
petitsuix
10
i
numeronobis
10
i
justforkix
10
p
i
dairiprodus
10
i
vitriolix
10
i
giveusabonus
10
i
nautilus
10
r
q
```

Output:

```
Command? name? height? Command? name? height? Command? name? height?
Command? justforkix is 10.00 cm tall
numeronobis is 10.00 cm tall
petitsuix is 10.00 cm tall
Command? name? height? Command? name? height? Command? name? height?
Command? name? height? Command? vitriolix is 10.00 cm tall
petitsuix is 10.00 cm tall
numeronobis is 10.00 cm tall
nautilus is 10.00 cm tall
justforkix is 10.00 cm tall
giveusabonus is 10.00 cm tall
dairiprodus is 10.00 cm tall
Command? Bye!
```