

# Programming in C

## Supervision 1

**Supervisor:** Joe Isaacs (josi2).

All work should be submitted in PDF form 36 hours before the supervision to the email [josi2@cam.ac.uk](mailto:josi2@cam.ac.uk). If you have any questions on the course please include these at the top of the supervision work and we can talk about them in the supervision.

Please attempt all exam questions *without a compiler* at first, then once you are happy with the solution then use a compiler to check for errors.

1. What is type widening and narrowing and how can they cause program errors.
2. How are variables defined at the top level (global scope) and what semantics does this have. What about within a block, also when are they allocated and deallocated.
3. Will `char i, j; for(i=0; i<10, j!=5; i++, j++) ; terminate?` If so, under what circumstances?
4. Write an implementation of bubble sort for a fixed array of integers. (An array of integers can be defined as `int i[] = {1, 2, 3, 4}`; the 2nd integer in an array can be printed using `printf("%d\n", i[1]);`.)
5. Modify your answer to (4) to sort characters into lexicographical order. (The 2nd character in a character array `i` can be printed using `printf("%c\n", i[1]);`.)
6. What is the difference between a function definition and declaration. Why would one want to declare a function and not define it?
7. What is the call scheme for functions in C? And how does this scheme work?
8. What arguments are allowed by the function definition `int power()`; in C vs in C++, and how do we achieve the C++ semantics in C?
9. What does `#include"abc.h"` do and why do we use it?
10. Describe the address space layout in C.
11. How can you prevent a header being included twice?
12. Define a macro `SWAP(t, x, y)` that exchanges two arguments of type `t`.
13. Does your macro work as expected for `SWAP(int, v[i++], w[f(x)])`?
14. Define a macro `SWAP(x, y)` that exchanges two arguments of the same type (e.g. `int` or `char`) without using a temporary.
15. <http://www.cl.cam.ac.uk/teaching/exams/pastpapers/y2007p3q4.pdf> Part a only. Using the C operator precedence table here [http://en.cppreference.com/w/c/language/operator\\_precedence](http://en.cppreference.com/w/c/language/operator_precedence)
16. <http://www.cl.cam.ac.uk/teaching/exams/pastpapers/y2007p11q3.pdf> Apart from (b) and for (c) using C (not C++).
17. What is array syntax short for?

18. What is the output of this program chunk and why?

```
int* a = 0x7fff5c2d1a9c;
char*b = (char*)a;
printf(" address of a+1: %p, address of b+1 %p",
      (void*)(a+1), (void*)(b+1));
```

given that `%p` is the format specifier for a pointer and `(void*)` will cause the address of the expression to be correctly formatted. Assume that `int` is 4 words.

19. What does a multi-dimensional array get compiled down to?
20. If `p` is a pointer, what does `p[-2]` mean? When is this legal?
21. Write a program `calc` which evaluates a reverse Polish expression given on the command line; for example `$ calc 2 3 4 + *` should print 14.
22. What is the use of the `const` keyword in C and what does each `const` mean in the statement `const int *const p`
23. Why may it be useful to `typedef` a `struct`

```
typedef struct llist {
    ...
} linkedlist
```

24. Implement a queue of ints with the following access methods in a header file and the implementation in a `.c` file.

```
void queue_init(queue* que, size_t size_hint);
void queue_destroy(queue* que);
int queue_dequeue(queue* que);
int queue_enqueue(queue* que, int value);
int queue_empty(queue* que);
```

The implementation of the queue is up to you. You could use a array or a linked list, what would be benefits being to using one over the other? Make sure multiple queues can be instantiated using this interface.

25. Populate your queue from a file using `fscanf`. The input in each case should be a list of whitespace delimited (separated) numbers.

```
1 2 100 23 -2
```

then write the numbers from to queue out to stdout.