The Open University of Sri Lanka
Faculty of Engineering Technology
Department of Mechanical Engineering

Study Programme	: Bachelor of Technology Honours in Engineering
Name of the Examination	: Final Examination
Course Code and Title	**: DMK3270/MEK3170 C Programming**
Academic Year	: 2019/2020
Date	: 27th September 2020
Time	: 0930-1230hrs
Duration	: **3 hours**

General Instructions

1. Read all instructions carefully before answering the questions.
2. This question paper consists of **Seven (7)** questions in **Five (5)** pages.
3. Answer any **Five (5)** questions only. All questions carry equal marks.
4. Answer for each question should commence from a new page.
5. Relevant charts/codes are provided.
6. This is a Closed Book Test **(CBT).**
7. Answers should be in clear handwriting.
8. Do not use a red color pen.

Q1

(a) Briefly explain the reason/s for using a "head file" in C programming. What are the functions of the following header files?

i. iostream.h
ii. math.h
iii. stdio.h

[6 marks]

(b) Briefly explain data types and variable types used in C programing.

[4 marks]

(c) What are the differences between the constants 7, '7', and "7"?

[6 marks]

(d) What is the difference between the constants 123 and "123"?

[4 marks]

Q2

(a) Consider the following program, written to test whether the representation of *int* on a particular architecture uses two's complement or not.

```
#include <limits.h>
#include <stdio.h>
int main(void) {
printf("%d\n", (INT_MAX + 1) < 0);
return 0;
}
```

Explain what this program is permitted to output, and why.

[5 marks]

(b) Consider the following C function, intended to receive a string as an argument and return a reversed string as a result. (The strlen function returns the number of characters in the string.)

```
#include <string.h>
#include <stdlib.h>
char *reverse(char *input) {
int len = strlen(input);
char output[len];
for (int i = 0; i < len; i++) {
output[len - i - 1] = input[i];
}
return output;
}
```

(i) Identify and state three bugs in this function.

[6 marks]

(ii) Write a correct version of this function.

[9 marks]

Q3

(a) In a C++ program, suppose there is a function with the following prototype.
void foo(MyClass x)
Suppose that this function is invoked in a call foo(z).

(i) What does C++ do when z is passed as an argument to foo?

[4 marks]

(ii) Explain briefly how MyClass class can be modified to raise compile-time errors when objects of type MyClass are passed as arguments.

[6 marks]

(iii) Why might you want to do this?

[4 marks]

(iv) In response, how should the type of foo be declared instead? Give a new function prototype for foo.

[6 marks]

Q4

In C, it is typical for APIs to expose functions that create values (such as fopen) and then subsequently delete them (such as fclose). For example, a C program might use files with code such as:

```
void bar(void) {
FILE *fp = fopen("example.txt","r");
baz(fp);
fclose(fp);
}
```

(a) What makes this style of resource management problems in C++? Your answer should explain what baz could do that creates a resource-management hazard.

[6 marks]

(b) Describe the preferred alternative to handling this kind of resource management issue in modern C++.

[6 marks]

(c) Define a C++ class that wraps the C file API to support C++ style resource management. You only need to show how to wrap the resource management calls (fopen and fclose), and may ignore the rest of the file API.

[8 marks]

Q5

Consider the following C program:

```
void swap(int x, int y) {
int temp = x;
x = y;
y = temp;
}
int main(int argc, char **argv) {
int x = 0;
int y = 1;
swap(x, y);
assert(x == 1);
return 0;
}
```

(a) Briefly explain the role of the assert statement and why this program will trigger an assert failure when executed. Supply two modified versions of the program that alter the swap function definition and, if necessary, its calls to avoid this assert failure. One version should be in C, and the other should use C++ language features.

[8 marks]

(b) Describe the address-space layout (highlighting four areas of memory) of a typical compiled x86 C program, and how each of these areas are used by C constructs.

[6 marks]

(c) Briefly explain what undefined behavior is in the C standard. Under what circumstance(s) would be calling the following C function result in undefined behavior?

```
int32_t divide(int32_t a, int32_t b)
{
return a / b;
}
```

[6 marks]

Q6

(a) Write a C function revbits(), which takes a single 8-bit char parameter and returns a char result by reversing the order of the bits in the char.

[4 marks]

(b) Write a C function revbytes() taking two parameters and returning no result. The first parameter is a pointer to memory containing n contiguous bytes (each of type char), and the second is the number of bytes. The function should have the side effect of reversing the order of the bits in the n contiguous bytes, seen as a bitstring of length 8n. For example, the first bit of the first char should be swapped with the last bit of the last char.

[6 marks]

(c) You have been assigned the following seemingly working C code, which processes files controlling the behavior of a system. You observe that, after obtaining several ERR_MALFORMED errors, subsequent calls to fopen fail due to too many files being open:

```
int process_file(char *name)
{ FILE *p = fopen(name, "r");
if (p == NULL) return ERR_NOTFOUND;
while (...)
{ ...
if (...) return ERR_MALFORMED;
process_one_option();
...
}
fclose(p);
return SUCCESS;
}
```

i. Explain how to fix the program using facilities in C.

[2 marks]

ii. Now suppose the function above was part of a system written in C++ (but still using the C file-processing commands such as fopen and fclose), and that process_one_option() might raise one or more exceptions. Using a class with a destructor, show how to fix the "too many files open" bug above.

[8 marks]

Q7

(a) Considering unspecified behavior in C.

i. Define what unspecified behavior means in the C standard and give two examples of such behavior.

[3 marks]

ii. Briefly explain why it is important to have unspecified behavior in the definition of the C language.

[1 mark]

(b) Compare and contrast the struct and union keywords in C, supplying an example of a situation where it would be more appropriate to use a union rather than a struct.

[4 marks]

(c) Explain the following C or C++ language concepts. You may find it helpful to use short code fragments or diagrams to illustrate your answer.

i. The virtual keyword used to qualify a C++ member function and its impact on generated code.

[4 marks]

ii. The role of the C preprocessor in the source-code compilation cycle, and why it is a useful tool for debugging.

[4 marks]

iii. Templated functions in C++, giving one benefit and one drawback of using them compared with using a void* function in C.

[4 marks]

-End-