University of Glasgow

PHYSICS 2T
C Programming under Linux
14:00–15:00, Tue 27th Apr 2021

Candidates should answer:

- **All three questions [20 marks each];**

---

1. Taking the following steps, write down some c-code to read in a data file **data.dat** consisting of two columns of 100 integer numbers:

a) Start your code with brief header comments and include any necessary header files. [3]

b) In the main function, create a file pointer and open an existing data file called **data.dat**. [3]

c) Declare a two-dimensional integer array (or two one-dimensional arrays) to contain the data. [1]

d) Loop over the data file and read in each line. [3]

Assume the code you have written above is included in a small software project as the file **main.c** and that you have been given a **Makefile** for that project as shown below:

```
1     CC=gcc
2     EXE=stats
3     OBJS= access.o util.o main.o
4
5     $(EXE): $(OBJS)
6           $(CC) $^ -o $@
7
8     %.o: %.c
9           $(CC) -c $<
10
11
12    .PHONY: test
13    test:
14          ./stats –h testportal.gla.ac.uk
```

Continued overleaf

Answer the following questions:

e) Explain what happens when the user runs the **make** command and specify what files are produced by the compiler.

[3]

f) Explain what the variables $^ and $@ mean on line 6 and expand all the variables on that line to show the compilation line that would be executed.

[3]

g) What changes need to be made to the above Makefile to add the following features:

i. Add an additional file **db.c** to the compilation of the stats executable. [2]

ii. Add a clean target that will remove any intermediate files. [2]

Continued overleaf

2. After examining the BASH script below, answer the following questions:

a) Describe what the four main parts of the script do (described by the line numbers below), paying close attention to any exit conditions or tests.

i. Lines 1 to 6 [2]
ii. Lines 8 to 15 [2]
iii. Lines 18 to 21 [2]
iv. Lines 23 to 25 [1]

b) Explain what "**`> /dev/null 2>&1`**" means, as used on line 11. [2]

c) Explain the difference between single quotes (') and double quotes (") in Bash string interpolation and why double quotes (") are required on line 19.

[2]

d) Modify the script to accept multiple files as arguments and to process each one accordingly.

[3]

e) Replace lines 11-15 with a function named **testCommand**, which takes two parameters, 1) the name of a command to test and 2) the exit code to be used if the the command is not present. You should write the function and also show how it would be called in place of lines 11-15.

[6]

```
1     #! /bin/bash
2
3     if [ "$#" -ne "1" ]; then
4         echo "USAGE: pullprint FILENAME"
5      exit 1
6     fi
7
8     GUID=gl23x
9     FN_PRINT=$1
10    P_URL=sc-spooler.campus.gla.ac.uk/PullPrint
10
11    which smbclient > /dev/null 2>&1
12    if [ "$?" -ne "0" ]; then
13        echo "ERROR: smbclient not found"
14        exit 2
15    fi
16
17
18    if [ ! -e "$FN_PRINT" ]; then
19        echo "ERROR: File ${FN_PRINT} does not exist"
20        exit 3
21    fi
22
23    smbclient -U campus\\${GUID} //${P_URL} \
24                                      -c "print ${FN_PRINT}"
25    exit 0
```

Continued overleaf

3. The C code below iteratively estimates the value of π according to the algorithm described in the header comments and stops when it reaches a precision determined by the precision variable.

a) Explain what the statement on line 9 does and why it is needed. [2]

b) Write down the changes needed for the program to read in the precision as a command line argument: i.e. the user would be able to type ./piCalc 0.00001 to run the program. [5]

c) The program below will use the same sequence of random numbers each time it is run. How could you randomize the starting point of the random number sequence? [4]

d) Explain what (float) does on lines 22 and 23 and why it is needed. [2]

e) Explain why the ++ operator must be used *before* the operand in line 31 but can be used *after the operands* in lines 19 and 27. [2]

f) The code below writes the result to the terminal. Write down the additional code necessary to write the estimate of π to a text file (show code to open, write-to, and close the file). [5]

```
1    /*Program piCalc.c written by David Britton - Jan 2020             *
2    *                                                                  *
3    * Calculate π using the fact that for a large number of random     *
4    * pairs (a,b) with values of a and b between 0 and 1, π is given   *
5    * by approximately four times the number of pairs that satisfy the *
6    * condition a*a + b*b <1 divided by the total number of pairs.     */
7
8
9   #include <stdio.h>
10  #include <math.h>  // fabs() function: (absolute value of a float)
11  #include <stdlib.h>
12
13   int main(){
14  /*Variable declarations*/
15   int trys = 0, hits = 0, iterations = 100000;
16   float pi, a, b, precison = 0.00001;
17
18   do {
19    for (int index = 0; index < iterations; index++)  {
20
21     /*Generate random numbers from 0 to 1*/
22     a = (float)rand()/RAND_MAX;
23     b = (float)rand()/RAND_MAX;
24
25     /*update hits if condition is fulfilled*/
26     if (a*a + b*b < 1)
27       hits++;
28    }             //end of for-loop
29
30  /*Calculate Pi and print result*/
31   pi = 4.0 * hits / (iterations * ++trys);
32   printf("Pi is approximately %f\n", pi);
33
34  } while( fabs(3.1415926 - pi) > precison);
35
36  return 0;
37 }
```

End of question paper