INFO1112 Final Exam 2020
Section 2: extended answers (60 Marks)

ENTER YOUR 9-digit STUDENT NUMBER IN THE BOX:

490095315

Each question in this exam begins at the top of a new page. We have included a box on the following page for you to type your answer. Do not enlarge the box onto the next page.

Please do not change the font or font size.

When you are finished answering all the questions and before the deadline for completing the exam, please save your submission both as a Word file **and as a PDF**.

Upload your **PDF file** to Canvas as indicated on the Canvas Exam Quiz section 2.

You may upload more than once and we will use the most recent version.

Suggestion: each of these questions involves code (shell or python). You can test your code on your local machine and then cut/paste your solution into the template.

Question 1 (10 marks). The following python program was supposed to start a child process and wait for it to finish, but it has bugs. Rewrite the program to fix the bugs, and put your answer in the box on the next page.

```
pid = os.fork()
if pid != 0:
     time.sleep(5)
     print ("exit with status 99.")
     os.exit(99)
elif pid == 99:
     print ("yikes! fork failed!")
     sys.exit(1)
else:
     print ("waiting for PID", pid)
     # missing line of code here
     print ("process ID was ")
     # missing line of code here
```

Put your answer in the box on the next page.

DO NOT ENTER MORE THAN A PAGE FOR YOUR ANSWER TO EACH QUESTION

```
import sys ,time, os

pid = os.fork()
try:
     if pid != 0:
          print ("waiting for PID", pid)
          child_pid, exit_code = os.wait()
          print ("process ID was {}".format(child_pid))
     else:
          time.sleep(5)
          print ("exit with status 99.")
          os.exit(99)
except OSError:
     print ("yikes! fork failed!")
     sys.exit(1)
```

DO NOT ENTER MORE THAN A PAGE FOR YOUR ANSWER TO EACH QUESTION

Question 2 (10 marks). Write a **shell** script called **srch** that takes two arguments: a directory name and regular expression. The script should search the file system starting at the given directory and print the full pathname of all files with a filename that matches the regular expression.
Example command:

```
srch mydir '\.py$'
```

This command will search the file system starting at "mydir" and print the full pathname of all the files with a name that ends in .py

Put your answer in the box on the next page.

```
# answer to Q2

for each_file in `find $1 -type f -name $2 | awk '.' '{print
$2"."$3}' `
do
     echo `pwd`|$each_file
done
```

DO NOT ENTER MORE THAN A PAGE FOR YOUR ANSWER TO EACH QUESTION

Question 3 (10 marks). Write a **python** script, srch.py, that accepts two arguments: a directory and regular expression. The script should search the file system starting at the directory. For each filename that matches the regular expression, your script should find the largest file and print its full pathname and size. You must not invoke shell commands from within your script.
Example:

```
srch.py mydir '\.py$'
```

Put your answer in the box on the next page.

```
#!/usr/bin/env python3
# answer to Q3

import sys, re, os

if (len(sys.argv) < 3):
     quit()

folder, aim = sys.argv[1], sys.argv[2]
file_size = {}

regex = re.compile('({})'.format(aim))

for root, dirs, files in os.walk(folder):
     for f in files:
          if regex.match(f):
               file_size[f] = os.path.getsize(f)

max_key = max(file_size, key=file_size.get)
print("{} : {}".format(max_key,  file_size[max_key]))
```

DO NOT ENTER MORE THAN A PAGE FOR YOUR ANSWER TO EACH QUESTION

Question 4 (10 marks). Word scramble is a puzzle where you are given a string of nine scrambled letters. You have to find a word that uses those exact nine letters. Write a python function called **solve** that uses the file of words /usr/share/dict/words to return the first word that solves the puzzle.

Example of the use of the function:

```
    answer = solve("nsaklbeat")
    print (answer)
```

The output should be:

```
beanstalk
```

Put your answer in the box on the next page.

```
#!/bin/env python3
# answer to Q4
def solve(word):
     with open("/usr/share/dict/words", "r") as f:
          words = [i.strip("\n") for i in f.readlines()]

     for each_word in words:
          if len(each_word) != len(word):
               continue

          check , aim = list(word), list(each_word)
          while check:
               if check[0] in aim:
                    if len(check) == 1 and len(aim ) == 1:
                         return each_word
                    check.remove(check[0])
                    aim.remove(check[0])
               else:
                    break

     return ""
```

DO NOT ENTER MORE THAN A PAGE FOR YOUR ANSWER TO EACH QUESTION

Question 5 (10 marks). Given the **solve** function from the previous question, write a program that implements a scramble **server.** It should open a network socket for localhost and port 7070, listen on the socket for a string of nine scrambled letters, solve it using the solve function, then return the result to the client program. It should then go back to listening for another word scramble.

Put your answer in the box on the next page.

```
#!/usr/bin/env python3
# answer to Q5 scramble server


import socket

serv = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serv.bind(('', 7070))
serv.listen(5)
while True:
     conn, addr = serv.connect()
     recv_data = conn.recv(1024).decode()
     words = solve(recv_data)
     conn.sendall(words.encode())
     conn.close()
```

DO NOT ENTER MORE THAN A PAGE FOR YOUR ANSWER TO EACH QUESTION

Question 6 (10 marks). Given the solve server from the previous question, write a program that implements a scramble **client.** It should prompt the user for a set of nine letters (check for errors), open a network socket to localhost and port 7070, send the string of scrambled letters, read the result and print it.

For example:

```
$ solve-client
scramble? nsaklbeat
beanstalk
```

Put your answer in the box on the next page.

```
#!/usr/bin/env python3
# answer to Q6 scramble client


import socket

words = input("Enter a nine-letter word:")
if len(words) != 9:
 quit()

client = socket.socket(socket.AF_INET ,socket.SOCK_STREAM)
client.connect(("localhost", 7070))
client.sendall(words.encode())
words = client.recv(1024)
print(words.decode())
client.close()
```

.

DO NOT ENTER MORE THAN A PAGE FOR YOUR ANSWER TO EACH QUESTION