Please fill in your Student Number and Name.

Student Number : _____

Name:

_____

_____

Student Number:

_____

University of Cape Town ~ Department of Computer Science

Computer Science 1015F ~ 2011

June Examination

** *SOLUTIONS* **

Question	Max	Mark	Moderator
1	10		
2	12		
3	18		
4	20		
5	10		
6	10		
7	20		
TOTAL	**100**		

Marks : 100

Time : 180 minutes

Instructions:

a) Answer all questions.

b) Write your answers in PEN in the spaces provided.

c) Show all calculations where applicable.

Question 1 [10]

(a) Explain clearly and concisely:

i. The **purpose** of an **interpreter**. [3]

A interpreter is a complex program that translates [1] a program written in a high-level language[1] into machine code [1].

ii. The **advantages** of using a **compiler** instead of an interpreter. [4]

An interpreter translates the high-level source program line-by-line into machine code, whereas a compiler does it all at once [1]. For a compiler, only the machine code is necessary to run a program [1]- an interpreter requires both the interpreter and the source code.[1] It is also faster to run compiled code. [1]

(b) Python has **two** basic numeric data types. What are they called and why is it it necessary to have two? [3]

int and float[1] Integer and floating point arithmetic is handled differently in the hardware. [1] Interger arithetic is faster and therefore it makes sense to use the integer arithmetic when possible. [1]

Question 2 [12]

Examine the module listed below:

```
# module1.py

def display():
    print("...")

def process(word):
    if(word=="$"):
        return(8)
    elif(word=="€"):
        return(10)
    elif(word=="£"):
        return(12)
    elif(word=="kr"):
        return(1.2)
    elif(word=="R"):
        return(1)
    else: return 0

def main(word1, word2):
    display()
    print(word1,word2,sep='',end="=")
    f= process(word1)
    if f:
        r=word2*f
        print("R",r,sep='')
    else:
        print("R?")

main("$",10)
main("kr",20)
main("R",15)
main("zl",50)
```

(a) From "module1.py" listed above**,** give an example of:

i. a variable [1]

Any of the variables used in the program: word, word1, word2,f , r etc.

ii. a literal [1]

Any of the strings ("$","R", "kr", "..." etc) or the number values (20,100 etc.) used in the program

iii. a numerical operator [1]

*

iv. a relational operator [1]

==

(b) Write down the **exact** output of this module when it is run in the Python interpreter (i.e. when you press "Run" in the Wing IDE). [6]

…

$10=R80

...

kr20=R24.0

...

R15=R15

...

zl50=R?

1 mark for all dotted lines, 1 mark for each other line of text correct (=4), 1 mark for correct formatting of all lines

(c) The built-in Python function `ord` is invoked in the Python interpreter as follows.

```
>>> ord("£")
163
```

Explain what this function does and the significance of the number returned (163). [2]

ord returns the Unicode value of the character passed to it. The Unicode value of the pound sign is 163: this is the numeric code used to represent the character.

Question 3 [18]

Examine the module below and answer the questions that follow.

```
# module2.py

def problem(first, second):
    first,second = first.upper(),second.upper()
    print(second)
    p=0
    for i in first:
        for j in second[p:]:
            p=p+1
            if i==j:
                print(i,end='')
                break
            print(' ',end='')
        else:
            print("\nThere is no",first,"in",second,"!")
            break
    print()

problem("nap","narcolepsy")
problem("nap","insomnia")
```

(a) Explain what the `break` statement does in Python. [2]

The break statement breaks out of the immediately enclosing loop – the loop ends.

(b) Explain in general terms what the function `problem` does. [2]

It determines whether the letters on one string can be found in order in another string. [1] .It prints out the second string and the first string below it, with the letters of the first string arranged to match their positions in the second string. If the string cannot be found, it prints out a message.

(c) Write down the **exact** output of this module when it is run in the Python interpreter (i.e. when you press "Run" in the Wing IDE). [6]

NARCOLEPSY
NA P
INSOMNIA
N A
There is no NAP in INSOMNIA !

(d) The function `problem` uses two nested **for** (definite) loops. Rewrite the function so that it **works exactly the same as before** but now uses two nested **while** (indefinite) loops. [8]

```
def problem2(first, second):
    first,second = first.upper(),second.upper()
    print(second)
```

```
i,j =0,0 #[2]
while(i<len(first)): #[1]
   while(j<len(second)): #[1]
      if first[i]==second[j]: #[1]
         print(first[i],end='')
         j=j+1 #[1]
         break
      j=j+1 #[1]
      print(' ',end='')
   else:
      print("\nThere is no",first,"in",second,"!")
      break
   i=i+1 #[1]
print()
```

Question 4 [20]

Answer the following questions based on the code below.

```
def read_file ():
   f = open ("infile.txt", "r")
   lines = f.readlines ()
   f.close ()
   return lines

def process (lines):
   store = {}
   num = 0
   while num < len(lines):
      if not (lines[num] in store):
         store[lines[num]] = 1
         num += 1
      else:
         del lines[num]

def write_file (lines):
   # write contents of list to outfile.txt

def main ():
   data = read_file ()
   process (data)
   write_file (data)

if __name__ == "__main__":
   main ()
```

(a) What is a list/array?. [2]

linear ordered sequence of data items associated with one variable name

(b) Explain clearly what the **process** function accomplishes. [2]

removes duplicates from list

(c) Explain what the purpose of the final 2 lines of the program is. [2]

execute main if the program has been executed directly (as opposed to executed as a module)

(d) In the process function, a while loop is used instead of a for loop. Why? [2]

because the size of the list varies over time (as items are removed)

(e) How many test cases are need to test the **read_file** function if one uses **path testing**? How many test cases are need to test the **read_file** function if one uses **statement coverage**? [2]

1

1

(f) Under what circumstances should a dictionary be used instead of a list in a program? [2]

when there is a mapping of keys to values and there is a need for fast lookup of keys

(g) Write code to strip off newline ("\n") characters from the end of each line in a list (named **data**) Assume that every line ends in a newline character. [3]

```
for d in range (len (data)): [1]
  data[d] = data[d][:-1] [2]
```

(h) Write the **write_file** function to write the contents of the list to a text file called "outfile.txt".[5]

```
def write_file (lines):
  f = open ("outfile.txt", "w") [1]
  for line in lines:[1]
     print (line, end="", file=f) [2]
  f.close () [1]
```

Question 5 [10]

(a) Sort the following numbers using the recursive **quicksort** algorithm (using the last element as the pivot). Clearly show each of the 3 partitioning steps of the algorithm. [3]

```
8 17 13 12 5 10 14 11
```

8 5 10 [11] 17 13 14 12

8 5 [10] [12] 17 13 14

[5] 8 13 [14] 17

5 8 10 11 12 13 14 17

(b) What is the average time complexity of the quicksort algorithm? [1]

O(n log n)

(c) Which algorithm is better: quicksort or selection sort? Why? [1]

quick. selection is O(n)

(d) Write a **recursive function** to calculate the power of a number (a^b). [5]

```
def power (a, b):[1]
  if b == 0: [1]
    return 1 [1]
  else:
    return a * [1] * power (a, b-1) [1]
```

Question 6 [10]

(a) Convert the decimal number 37 to binary, showing your working. [2]

37/2 = 18 rem 1

18/2 = 9 rem 0

9/2 = 4 rem 1

4/2 = 2 rem 0

2/2 = 1 rem 0

1/2 = 0 rem 1 answer: 100101

(b) Convert the hexadecimal number 456 to octal, showing your working. [2]

010|001|010|110

answer: 2126

(c) Calculate 12 - 2 using 8-bit 2's complement binary representation, showing all your working. (i.e. convert the numbers to 8-bit 2's complement before adding). Leave your final answer in 2's complement form. [3]

00001100 – 00000010

= 00001100 + 2s(00000010)

= 00001100 + 11111110

= 00001010

(d) Find the value represented by the floating point number written below using IEEE754 representation (bias of 127 added to the exponent; first bit is the sign bit, next 8 bits are the biased exponent, rightmost 23 bits are the significand). Show all your working. [3]

`0 01111101 00100000000000000000000`

sign implies positive value

biased exponent is 125 so actual exponent is 125 – 127 = -2

mantissa is 1.001 in binary which is 1.125 in decimal

*so number is 1.125 * 2^-2 = 0.28125*

Question 7 [20]

SRC Elections

Local elections are now over, but soon students will be called upon to vote in the Student Representative Council (SRC) elections. In these elections, each student **may vote multiple times**, for up to 13 candidates. The SRC has decided to use computers instead of paper ballots.

You have been hired by the SRC to write a simple software application to be used on these computers.

Your program must ask each user for a list of candidate names and then update a set of counters. The counters can be stored in a list or dictionary in memory.

(a) Write Python code to allow a single user to enter a sequence of up to 13 candidate names and update the global vote counters after each user votes. Do not include error checking. [8]

```
name = input ("Enter a name")
number_of_names = 0
while number_of_names < 13 and name != "done":
  if name in votes:
    votes[name] = votes[name] + 1
  else:
    votes[name] = 1
  number_of_names += 1
  if number_of_names < 13:
    name = input ("Enter a name:")
```

(b) Explain 2 ways in which you can use text files to make your program more robust, in the face of a possible power failure. [4]

store vote counters after each voter is done.

store all votes in a file as votes are placed

(c) Describe the algorithm you could use to check that the names of candidates are not repeated in the set of votes for a single voter. [4]

create an empty dictionary before names are entered. for each entered name, check if it is in the dictionary. if not, add it, if yes, it is an error.

(d) Describe one possible useful set of equivalence classes and boundary values that can be used to test the program. [4]

equivalence classes for different numbers of votes

boundary values at 0 and 13 votes