Model Answers for Module BUCI007H4

Summer

Birkbeck

(University of London)

BSc/FD EXAMINATION

Department of Computer Science and Information Systems

Introduction to Programming (BUCI007H4)

CREDIT VALUE: 15 credits

Summer 2020

On Line Version

Date of examination: TUESDAY 26th MAY 2020
Duration of paper: 14:00 – 18:00

The programming language is Python.

There are **ten** questions in this paper.

Answer all **ten** questions.

Each question carries **10** marks in total.

The examination is open book.

No supplementary material is provided.

This paper is not prior disclosed.

The quote marks for strings are in this style "string" or 'string'.

The marking scheme favours the understanding of the material, the correctness of any working and the quality of argument.

1. (a) Write down the result of running the following Python code. Justify your answers. (**5** marks)

   i) `print(len('9yth0\\\n'))`

   ii) `print('3\'4')`

   iii) `len("4-1") = 3`

   iv) `print(3*"ab"*2)`

   v) `print("He said, 'No!'")`

   (b) Which of the following are rejected by the compiler as names of variables? Specify the reason why they are accepted or rejected. (**5** marks)

   i) `6packs`

   ii) `state-of-the-art`

   iii) `While`

   iv) `_30` # One underscore, followed by 30

   v) `true`

<u>Answer</u>:

(a) *i*) 7

ii) 3'4

iii) error

iv) abababababab

v) He said, 'No!'

(b) Variable names must begin with a letter or underscore. Other characters can be letters, numbers or underscores. Keywords cannot be variable names. *1 mark each.*

(c) *i*) not accepted, because it starts with a number;

ii) not accepted, because of the -

iii) accepted

iv) accepted

v) accepted

1 mark each.

2. Consider the following code. (**10** marks)

```
a = -2
b = 7
c = -5
d = 5
e = 3.5
a = b ** (a - c) - a * b
b = abs(-89 // b + 2) / 5
c = sqrt(c - 3 * -7) ** 4
d = 1000 % (d - 3) ** 3 ** 2
e = min(max(0.99, -3, 2E-2), max(-6, abs(-45/e) ), max(-3, -1))
```

What are the values of a, b, c, d and e after the above code is executed? Write down the step-by-step results of all the calculations. A correct answer without calculation details will not be awarded any marks.

Answer:

i) a = 357

$$\begin{aligned} a &= 7**(-2-(-5))-(-2)*7 \\ &= 7**3-(-14) \\ &= 343+14 \\ &= 357 \end{aligned}$$

ii) b = 2.2

$$\begin{aligned} b &= abs(-89//b+2)/5 \\ &= abs(-11)/5 \\ &= 11/5 \\ &= 2.2 \end{aligned}$$

iii) c = 256.0

$$
\begin{aligned}
c &= sqrt(c - 3 * -7) ** 4 \\
&= sqrt(c + 21) ** 4 \\
&= sqrt(16) ** 4 \\
&= 4 ** 4 \\
&= 256.0
\end{aligned}
$$

iv) d = 488

$$
\begin{aligned}
d &= 1000\%(d - 3) **3 **2 \\
&= 1000 \; \% \; 2 **9 \\
&= 1000 \; \% \; 512 \\
&= 488
\end{aligned}
$$

v) e = -1

$$
\begin{aligned}
e &= min(max(0.99, -3, 2E - 2), max(-6, abs(-45/e)), max(-3, -1)) \\
&= min(max(0.99, -3, 0.02), max(-6, abs(-12.86)), max(-3, -1)) \\
&= min(0.99, 12.86, -1) \\
&= -1
\end{aligned}
$$

2 marks each

3. (a) Consider the following code.

```
def num_func (number):
    temp = number
    rev = 0
    while number > 0:
        dig = number % 10
        rev = rev * 10 + dig
        number = number //10
    if temp == rev:
        return True
    else:
        return False
```

Explain what num_func does. In other words, when it will return True and when it will return False? (**4** marks)

(b) Write a Python program to find a three-digit number x, where num_func(x) will return True and when adding 2020 to x, the resulting number will also get a True from the num_func function. You should provide both the Python program and the number x in your answer. (**6** marks)

Answer:

(a) This function returns True if the parameter is a palindrome. A palindrome is a number that remains the same when read backwards.

(b)

```
for i in range(100, 1000):
    if num_func(i) and num_func(i+2020):
        print(i, i+2020)
```

$x = 202$ and $x + 2020 = 2222$

2 mark for the correct form loop condition, 2 marks for the correct form of if structure, 2 marks for the correct returns.

4. (a) At the self-checkout, a user is about to purchase a product. Write Python code to prompt the user with three strings: (**6** marks)

"Please enter the name of the product:"
"Please enter the price of the product:"
"Please enter the quantity of the product:"

Give reasonable variable names to each of the inputs. If the item name is "cigarette" or "alcohol", the age of the customer will be asked. If customers are allowed to purchase the item, print out the result in the following form:

The total price to purchase product XXX is £YYY.

Otherwise print out a message

"You should be at least 18 years old to purchase the item."

(b) Define a function first_last that takes in two non-empty strings as parameters. It repeats the first character of the first parameter for n times, where n is the length of the first parameter. It then repeats the last character of the second parameter for m times, where m is the length of the second parameter. The function returns the concatenation of the two results. For example,

first_last("Hello", "bye?") returns a string "HHHHH????". (**2** marks)

(c) Write down four different ways to get the underlined character "a" in the following list. You may use positive and negative indices.

furniture = ["Table", "Chair", "Bed", "Sofa", "Wardrobe"] (**2** marks)

Answer:

(a)

```
name = input("Please enter the name of the product:")
price = float(input("Please enter the price of the product:"))
quantity = int(input("Please enter the quantity of the product:"))
total = price * quantity
if name == "cigarette" or name == "alcohol" :
   age= int(input("Please enter your age:"))
   if age < 18 :
      print("You should be at least 18 years old to purchase the item.")
   else :
      print("The total price to purchase product", name, "is £" + str(total) +".")
else :
   print("The total price to purchase product", name, "is £" + str(total) +".")
```

2 marks for the correct form of the three input functions, 2 marks for the correct form of if/else structure, 1 mark for the correct result, 1 mark for the correct output. Any reasonable version accepted.

(b)

```
def first_last(string1, string2):
    return string1[0] * len(string1) + string2[-1]*len(string2)
```

one mark for each line

(c) furniture[1][2], furniture[-4][2], furniture[1][-3], furniture[-4][-3]

5. products is a list. Each member of products is another list specifying the name, price, quantity and discount of an individual product. For example,

```
products = [ ["Water bottle", 4.99, 200, 30],
             ["spoon", 1.49, 153, 15],
             ["Dessert plate", 12.99, 1009, 12.5] ]
```

Product names are of string type, prices and discount are of float type, and quantity is of int type. The longest name in products has is "Dessert plate", the highest price in price is 230.99, the largest quantity is 2400 and the highest discountis 50.0.

Write down a Python program to print all the product information in the following form:

```
Water bottle _|_0004.99_|__200_|_30.0%
Spoon _________|_0001.49_|__153_|_15.0%
Dessert plate_|_0012.99_|_1009_|_12.5%
```

_ indicates one space. Leave one space before and after the column separator |. (**10** marks)

Answer:

```
products = [ ["Water bottle", 4.99, 200, 30],["spoon", 1.49, 153, 15], ["Dessert plate", 12.99, 1009, 12.5]]
for prod in products:
    print("%-14s | %07.2f | %4d | %4.1f%%" % (prod[0], prod[1], prod[2],prod[3]) )
```

Correct format specifier 4 marks, correct application of the format specifier 2 marks, correct for loop 2 marks. Overall correct 2 marks. One mark deducted for each error.

6. (a) Let a, b, c be variables such that a has the value False, b has the value True and c has the value True. Use the truth tables for and and or to evaluate the expression

(a or b) and ((b and c) or c)

Describe the calculation in detail. (**4** marks)

(b) Evaluate the following expressions in which x has the value 5 and s is the string with value "123". In each case state whether the result is True or False. Justify your answers.

i) 3 ! = 8//4

ii) 1 < x < 3

iii) s = = "12"

iv) s[-1] > "1"

(**2** marks)

(c) Use the truth table for or to show that True or y can be evaluated even if the truth value of y is unknown. Similarly, use the truth table for and to show that False and z can be evaluated even if the truth value for z is unknown. Justify your answers.

(**4** marks)

Answer:

(a) The variable a has the value False and b has the value True. The truth table for or shows that (a or b) has the value True. The variables b and c both have the value True. The truth table for and shows that (b and c) has the value True. It follows that the expression in question reduces to (True and (True or c)). The truth table for or shows that (True or c) has the value True, thus the expression reduces to (True and True), which has the value True. 4 marks.

(b) i) True. The value of 8//4 is 2 and the value of 3 != 2 is True; ii) False. The expression is equivalent to $1 < x$ and $x < 3$. The expression $x < 3$ is False; iii) False. The string s has the value "123" and "123" == "12" is False; iv) True. s[-1] has the value "3" and it is True that $"3" > "1"$. 2 marks. 1/2 mark for each correct item.

(c) If y has the value True, then (True or y) has the value True. Similarly, if y has the value False then (True or y) has the value True. It follows that (True or y) has the value True in all possible cases. Similarly, (False and z) has the value False if z has the value False and it has the value False if z has the value True, i.e. (False and z) has the value False in all possible cases. (4 marks).

7. (a) Consider the two following snippets of Python code.

```
# code 1
if flag :
   print(True)
else :
   print(False)
   print("check if statement")


# code 2
if flag :
   print(True)
else :
   print(False)
print("check if statement")
```

i) Describe the calculations that are carried out if code 1 is executed with flag equal to True.

ii) Describe the calculations that are carried out if code 1 is executed with flag equal to False.

iii) Describe the calculations that are carried out if code 2 is executed with flag equal to True.

iv) Describe the calculations that are carried out if code 2 is executed with flag equal to False.

(**4** marks)

(b) In an elevator simulation the variables floor and maxFloor have integer values. The following values of floor are allowed: any integer in the range 1 to maxFloor inclusive, with 13 and 14 both omitted. Define a function called check that takes floor and maxFloor as arguments and returns True if floor has an allowed value. In all other cases check returns False. (**4** marks)

(c) Rewrite the function check to obtain a new function check1 with maxFloor as the only argument. The function check1 prompts for the value of floor using the function input. Once the value of floor is obtained, check1 makes a function call to check and returns the result of the function call. (**2** marks)

Answer:

(a) Code 1, flag == True: the first block in the if statement is executed, i.e. True is printed. Code 1, flag == False: the second block in the if statement is executed, i.e. False and "check if statement" are printed on consecutive lines. Code 2, flag ==

True: the first block in the if statement is executed, i.e. True is printed, then "check if statement" is printed on the next line. Code 2, flag = = False: the second block in the if statement is executed, i.e. False is printed. Then "check if statement" is printed on the next line. (4 marks, one mark each)

(b)

```
def check(floor, maxFloor) :
    if floor <= 0 or floor > maxFloor :
        return False
    if floor = = 13 or floor = = 14:
        return False
    return True
```

4 marks. One mark deducted for each error.

(c)

```
def check1(maxFloor) :
    floor = int(input("Please input floor: "))
    return check(floor, maxFloor)
```

2 marks. One mark deducted for each error.

8. (a) State the sequence of values of n that is printed when the following code is executed in a correct Python program.

```
n = 6
for i in range(4) :
   if n%3 == 0 :
      n = n//3
   else :
      n = 2*n+1
   print(n)
```

(**4** marks)

(b) Write a function f that takes a list L of integers as a parameter. The function f returns the largest value of the integer i such that L[i] equals L[i+2]. If no such integer exists, then f returns -1. (**6** marks)

Answer:

(a) 2, 5, 11, 23 (4 marks). One mark for each correct number.

(b)

```
def f(L):
   j = -1
   while i < len(L)-2 :
      if L[i] == L[i+2] :
         j = i
      i = i+1
   return j
```

Six marks. One mark deducted for each error.

9. (a) Consider the following function.

```
def main(n) :                # line 1
    sum = 0                  # line 2
    i = 0                    # line 3
    while i < n :            # line 4
        sum = sum+i          # line 5
        i = i+1              # line 6
        print(sum)           # line 7
```

Identify the local variables in main and describe the scope of each one. (**6** marks)

(b) State what is printed if the function main is called with the argument 2. Justify your answer. (**4** marks)

Answer:

(a) The local variables are sum (line 2 to line 7) and i (line 3 to line 7). 6 marks, 3 marks for each variable and scope.

(b) 0,1. The variables sum and i are assigned the value 0. The value of i < n is True, thus the while loop is entered. Then sum = sum+i = 0+0 = 0. Next, i is assigned the value 1 and 0 is printed. The value of i < n is True, thus the while loop is entered. Then sum = sum+i = 1. Next i is assigned the value 2 and 1 is printed. The value of i < n is False, thus the while loop is exited, and the function call is complete.

10. (a) The list ls is defined by ls = [6, 3, 1]. State all of the values of the index i such that ls[i] can be evaluated without error. (**2** marks)

(b) Evaluate ls[-ln[ls]]. (**2** marks)

(c) Let w be a list of length 3 such that each element of w is a list of integers of length 3. The elements w[0][0], w[1][1], w[2][2] form the diagonal. Write a function g that takes w as a parameter and that returns the sum of the elements of w not on the diagonal. (**6** marks)

Answer:

(a) -3, -2, -1, 0, 1, 2. (2 marks).

(b) 6 (2 marks).

(c)

```
def g(w):
   sum = 0
   for i in range(3) :
      for j in range(3) :
         if i != j :
            sum += w[i][j]
   return sum
```

Any reasonable answer accepted (6 marks).