

Sample Exam Questions

Final Exam – Part 2 (Writing Code)

تنبيه!

- هذه الأسئلة عبارة عن عينة فقط توضح طريقة أسئلة الاختبار النهائي - الجزء الثاني (كتابة البرامج) لمقرر البرمجة وحل المشكلات (CPIT-110).
- هذه الأسئلة لا يُعتمد عليها للمذاكرة.
- قد لا تشمل هذه الأسئلة جميع المواضيع المقررة للاختبار.
- هذه الأسئلة مناسبة للمراجعة بعد الانتهاء من مذاكرة وتطبيق المواضيع المقررة للاختبار.
- حلول الأسئلة مرفقة نهاية صفحات هذا الملف.
- الاختبار شامل جميع ما تم دراسته سابقاً من Chapter 1 إلى Chapter 6.
- هذا النموذج به أسئلة الباب السادس فقط، ونرجو مراجعة جميع نماذج الاختبارات السابقة للأبواب السابقة.

أسئلة الأبواب السابقة:

Chapter 1, 2, 3 :Sample Exam Questions Mid-Term Exam 1 – Part 2 (Writing Code)

Chapter 4, 5 :Sample Exam Questions Mid-Term Exam 2 – Part 2 (Writing Code)

<https://csu.kau.edu.sa/Pages-cpit110-test-questions.aspx>

Program 1

Programming Exercises (6.5)

Write the following function to display three numbers in increasing order:

Function Header	<code>def displaySortedNumbers(num1, num2, num3)</code>
Parameters	The three parameters (num1 , num2 , num3) are numbers.
Return Value	None

Write a test program that prompts the user to enter three numbers and invokes the function to display them in increasing order.

Here are some sample runs:



```
Enter three integers: 3, 2.4, 5 <enter>  
The sorted numbers are 2.4 3 5
```



```
Enter three integers: 100, 300, 200 <enter>  
The sorted numbers are 100 200 300
```

Solution

Program_1.py

```
1 def main():
2     num1, num2, num3 = eval(input("Enter three integers: "))
3
4     # Invoke the displaySortedNumbers method to display the
5     # numbers in increasing order
6     displaySortedNumbers(num1, num2, num3)
7
8
9 def displaySortedNumbers(num1, num2, num3):
10    # Write the code to implement this method
11    if num1 > num2:
12        num1, num2 = num2, num1
13
14    if num2 > num3:
15        num2, num3 = num3, num2
16
17    if num1 > num2:
18        num1, num2 = num2, num1
19
20    print("The sorted numbers are", num1, num2, num3)
21
22
23 # Call the main function
24 main()
```

Program 2

Programming Exercises (6.13)

Write a function to compute the following series:

$$m(i) = \frac{1}{2} + \frac{2}{3} + \dots + \frac{i}{i+1}$$

Function Header	def m(i)
Parameters	The parameter (i) is a number, and it represents the variable of the above equation.
Return Value	The result of the above equation. Note: This function does not print anything.

Write a test program that displays the following table:

i	m(i)
1	0.5000
2	1.1667
3	1.9167
...	
18	15.4523
19	16.4023
20	17.3546

Solution

Program_2.py

```
1 def main():
2     print(format("i", "<15s"), format("m(i)", "<20s"))
3     for i in range(1, 20 + 1):
4         print(format(i, "<15d"), format(m(i), "<20.4f"))
5
6 def m(n):
7     sum = 0
8
9     for i in range(1, n + 1):
10        sum += i / (i + 1)
11
12    return sum
13
14
15 main()
```

Program 3

Programming Exercises (6.17)

Create the following two functions:

Function Header	def isValid(side1, side2, side3):
Parameters	The three parameters (side1, side2, side3) are numbers.
Return Value	It returns True if the sum of any two sides is greater than the third side. Otherwise, it returns False. Note: This function does not print anything.

Function Header	def area(side1, side2, side3):
Parameters	The three parameters (side1, side2, side3) are numbers.
Return Value	It returns the area of the triangle using the following formula: $area = \sqrt{p \times (p - side1) \times (p - side2) \times (p - side3)}$ Where p is half the perimeter: $p = \frac{side1 + side2 + side3}{2}$ Note: This function does not print anything.

Write a test program that reads three sides for a triangle and computes the area if the input is valid. Otherwise, it displays that the input is invalid:



```
Enter three sides in double: 1, 3, 1 <enter>
Input is invalid
```



```
Enter three sides in double: 1, 1, 1 <enter>
The area of the triangle is 0.4330127018922193
```

Solution

Program_3.py

```
1 import math
2
3 def main():
4     edge1, edge2, edge3 = eval(input("Enter three sides in double: "))
5
6     if isValid(edge1, edge2, edge3):
7         print("The area of the triangle is", area(edge1, edge2, edge3))
8     else:
9         print("Input is invalid")
10
11 # Returns true if the sum of any two sides is
12 # greater than the third side.
13 def isValid(side1, side2, side3):
14     return (side1 + side2 > side3) and \
15     (side1 + side3 > side2) and (side2 + side3 > side1)
16
17 # Returns the area of the triangle.
18 def area(side1, side2, side3):
19     p = (side1 + side2 + side3) / 2
20     return math.sqrt(p * (p - side1) * (p - side2) * (p - side3))
21
22 main()
```

Program 4

Programming Exercises (6.6)

Write a function to display a pattern as follows:



```


        1
      2 1
    3 2 1
  ...
n n-1 ... 3 2 1

```

Function Header	def displayPattern(n):
Parameters	The parameter (n) is a number.
Return Value	None. It displays the above pattern.

Write a test program that prompts the user to enter a number n and invokes displayPattern(n) to display the pattern.

Here is a sample run:



```

Enter line number: 6 <enter>
        1
      2 1
    3 2 1
  4 3 2 1
5 4 3 2 1
6 5 4 3 2 1

```


Solution

Program_4.py

```
1 def main():
2     lineNumber = eval(input("Enter line number: "))
3     displayPattern(lineNumber);
4
5
6 def displayPattern(n):
7     for row in range(1, n + 1):
8         # Print spaces
9         for i in range(row, n):
10            print("  ", end="")
11
12        # Print numbers
13        for i in range(row, 0, -1):
14            print(format(i, "3d"), end="")
15
16        print()
17
18
19 main()
```

Program 5

Programming Exercises (6.18)

Write a function that displays an n-by-n matrix using the following header:

Function Header	def printMatrix(n):
Parameters	The parameter (n) is a number.
Return Value	None. It displays an n-by-n matrix.

Each element is 0 or 1, which is generated randomly. Write a test program that prompts the user to enter n and displays an n-by-n matrix.

Here are sample runs:

```
Enter n: 3 <enter>
```

```
1 0 0
0 1 1
1 1 1
```

```
Enter n: 10 <enter>
```

```
0 1 0 0 0 1 0 0 1 0
1 0 1 0 0 0 1 1 1 0
1 0 1 0 0 0 0 0 0 0
0 0 1 1 0 0 0 1 0 0
0 0 1 1 0 0 1 1 1 0
0 1 1 0 1 0 1 1 0 0
0 0 0 0 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 0 0 0 1 0
0 0 1 1 0 0 0 0 0 1
```

Solution

Program_5.py

```
1 import random
2
3 def main():
4     n = eval(input("Enter n: "))
5     printMatrix(n)
6
7
8 def printMatrix(n):
9     for i in range(1, n + 1):
10        for j in range(1, n + 1):
11            print(random.randint(0, 1), end=" ")
12
13        print()
14
15 main()
```

Program 6

Programming Exercises (6.8)

Write the following two functions:

Function Header	<code>def celsiusToFahrenheit(celsius):</code>
Parameters	The parameter (celsius) is a number.
Return Value	It returns the degree in Fahrenheit by using the following formula: $fahrenheit = (9 / 5) * celsius + 32$

Function Header	<code>def fahrenheitToCelsius(fahrenheit):</code>
Parameters	The parameter (fahrenheit) is a number.
Return Value	It returns the degree in Celsius by using the following formula: $celsius = (5 / 9) * (fahrenheit - 32)$

Write a test program that invokes these functions to display the following tables:

Celsius	Fahrenheit		Fahrenheit	Celsius
40	104.00		120	48.89
39	102.20		110	43.33
38	100.40		100	37.78
37	98.60		90	32.22
36	96.80		80	26.67
35	95.00		70	21.11
34	93.20		60	15.56
33	91.40		50	10.00
32	89.60		40	4.44
31	87.80		30	-1.11

Solution

Program_6.py

```
1 def main():
2     print(format("Celsius", "<15s"),
3           format("Fahrenheit", "<15s"), " | ",
4           format("Fahrenheit", "<15s"),
5           format("Celsius", "<15s"))
6     print("-----")
7
8     celsius = 40
9     fahrenheit = 120
10    i = 1
11
12    while i <= 10:
13        print(format(celsius, "<15d"),
14              format(celsiusToFahrenheit(celsius), "<15.2f"), " | ",
15              format(fahrenheit, "<15d"),
16              format(fahrenheitToCelsius(fahrenheit), "<15.2f"))
17        celsius -= 1
18        fahrenheit -= 10
19        i += 1
20
21    # Converts from Celsius to Fahrenheit
22    def celsiusToFahrenheit(celsius):
23        return (9.0 / 5.0) * celsius + 32
24
25    # Converts from Fahrenheit to Celsius
26    def fahrenheitToCelsius(fahrenheit):
27        return (5.0 / 9) * (fahrenheit - 32)
28
29    main()
```

Program 7


Programming Exercises (5.10) Modified

Write the following function:

Function Header	def getInputs(n):
Parameters	The parameter (n) represents the student number.
Return Value	First, this function should ask the user to input the student's name and score. After that, this function should return the entered name and score. (it returns two values)

Write a program that prompts the user to enter the number of students, and then the program should ask the user to enter each student's score by invoking the **getInputs** function. At the end of the input process, you program should displays the highest score.


Here are some sample runs:



```

Enter the number of students: 5 <enter>
Enter student #1 name: Ahmad <enter>
Enter student #1 score: 95.2 <enter>
Enter student #2 name: Omar <enter>
Enter student #2 score: 93.5 <enter>
Enter student #3 name: Jamal <enter>
Enter student #3 score: 95.5 <enter>
Enter student #4 name: Yeaser <enter>
Enter student #4 score: 80.5 <enter>
Enter student #5 name: Bander <enter>
Enter student #5 score: 66 <enter>

Top student Jamal's score is 95.5
    
```



```

Enter the number of students: 2 <enter>
Enter student #1 name: Ahmad <enter>
Enter student #1 score: 95.2 <enter>
Enter student #2 name: Omar <enter>
Enter student #2 score: 93.5 <enter>

Top student Ahmad's score is 95.2
    
```

Solution

Program_7.py

```
1 def getInputs(n):
2     name = input("Enter student #" + str(n) + " name: ")
3     score = eval(input("Enter student #" + str(n) + " score: "))
4     return name, score
5
6 # Prompt the user to enter the number of students
7 numOfStudents = eval(input("Enter the number of students: "))
8
9 # Initialize the variables for the highest score
10 student1, score1 = getInputs(1)
11
12 # Get students scores
13 for i in range(numOfStudents - 1):
14     student, score = getInputs(i + 2)
15
16     # Check the score of the entered student with the highest score
17     if score > score1:
18         student1 = student
19         score1 = score
20
21 # Display the output
22 print() # Empty line
23 print("Top student " + student1 + "'s score is " + str(score1))
```

Program 8

Programming Exercises (5.1) Modified


Write the following function:

Function Header	def getAverage(total, count):
Parameters	The parameters (total, count) are numbers.
Return Value	It returns the average of the numbers.


Write a program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (**not counting zeros**). Your program **ends with the input 0**.

Note: you should invoke the **getAverage** function to calculate the average of the numbers.

Here are some sample runs:



```
Enter an integer, the input ends if it is 0: 5 <enter>
Enter an integer, the input ends if it is 0: 8 <enter>
Enter an integer, the input ends if it is 0: -1 <enter>
Enter an integer, the input ends if it is 0: 2 <enter>
Enter an integer, the input ends if it is 0: 0 <enter>
The number of positives is 3
The number of negatives is 1
The total is 14
The average is 3.5
```



```
Enter an integer, the input ends if it is 0: 0 <enter>
No numbers are entered except 0
```


Solution

Program_8.py

```
1 def getAverage(total, count):
2     return total / count
3
4 # Define two variables to be counters for positive and negative numbers
5 countPositive = 0
6 countNegative = 0
7 # Define a variable to be a counter for all numbers regardless of +,-
8 count = 0
9 # Define a variable to store the sum of all entered numbers
10 total = 0
11
12 # Ask the user to enter a number
13 num = eval(input(
14     "Enter an integer, the input ends if it is 0: "))
15
16 # Define a while loop and set its condition to number != 0
17 # (while number is not equal to 0)
18 while num != 0:
19     # Check the number to determine if its positive or negative
20     if num > 0:
21         countPositive += 1
22     elif num < 0:
23         countNegative += 1
24
25     # Add number to total
26     total += num
27     # Increase count by one
28     count += 1
29
30     # Ask the user to enter the next number
31     num = eval(input("Enter an integer, the input ends if it is 0: "))
32
33 # Check count to determine if the user entered any number except 0,
34 # and then display the result
35 if count == 0:
36     print("No numbers are entered except 0")
37 else:
38     # Calculate the average
39     average = getAverage(total, count)
40
41     print("The number of positives is", countPositive)
42     print("The number of negatives is", countNegative)
43     print("The total is", total)
44     print("The average is", average)
```

Program 9

Chapter 5 – Program 4 (Modified)

Write the following function that randomly generates a subtraction question and asks the user for the answer to the question:

Function Header	def ask(from_n = 1, to_n = 100):
Parameters	The parameters (from_n, to_n) are numbers, and they represent the range of the random numbers that will be generated (inclusive).
Return Value	It returns True if the answer to the question is correct. Otherwise, it returns False .

Write a program to randomly generate subtraction questions and ask the user for the answer to each by **invoking** the **ask** function, and **let the user decide** whether to advance to the next question. Count how many the user got correct, and display the total time spent, by the user, answering the questions.

Note: if the first generated number is **less than** the second generated number, your program should swap their values.

Note: at the end of the program, display the time that the user is taken to answer all questions.

Here is a sample run:

```

What is 6 - 1? 5 <enter>
You are correct!
Enter Y to continue and N to quit: Y <enter>
What is 8 - 0? 6 <enter>
Your answer is wrong.
8 - 0 should be 8
Enter Y to continue and N to quit: Y <enter>
What is 8 - 3? 5 <enter>
You are correct!
Enter Y to continue and N to quit: N <enter>
Correct count is 2 out of 3
Test time is 24 seconds

```

Solution

Program_9.py

```
1 import random
2 import time
3
4 def ask(from_n = 1, to_n = 10):
5     # 1. Generate two random single-digit integers
6     number1 = random.randint(from_n, to_n)
7     number2 = random.randint(from_n, to_n)
8
9     # 2. If number1 < number2, swap number1 with number2
10    if number1 < number2:
11        number1, number2 = number2, number1
12
13    # 3. Prompt the student to answer "what is number1 - number2?"
14    answer = eval(input("What is " + str(number1) + " - " +
15        str(number2) + "? "))
16
17    isCorrect = number1 - number2 == answer
18    # 5. Grade the answer and display the result
19    if isCorrect:
20        print("You are correct!")
21    else:
22        print("Your answer is wrong.\n", number1, "-",
23            number2, "should be", (number1 - number2))
24
25    return isCorrect
26
27 correctCount = 0 # Count the number of correct answers
28 count = 0 # Count the number of questions
29
30 startTime = time.time() # Get start time
31
32 continueLoop = 'Y' # User confirmation flag
33 while continueLoop == 'Y':
34
35     # Ask a question, and increase correctCount if the answer is correct
36     correctCount += 1 if ask() else 0
37
38     # Increase the count
39     count += 1
40
41     # Prompt the user for confirmation
42     continueLoop = input("Enter Y to continue and N to quit: ")
43     print() # To insert a new line
44
45 endTime = time.time() # Get end time
46 testTime = int(endTime - startTime) # Get test time
47 print("Correct count is", correctCount, "out of",
48     count, "\nTest time is", testTime, "seconds")
```