Programming Thinking and Method Final Exam Key (A)

1 Multiple Choices (28 marks)

(Each question is worth 2 points)

1) A	2) B	3) C	4) D	5) D	6) B	7) A
8) C	9) C	10) B	11) A	12) C	13) D	14) A

2 Identification of Error Positions and Reasons in the Program (10 marks)

(Each answer is worth 2 marks)

Error Position (Statement)	Error Reason	Error Type
In the first line of the function body	Missing “longest = 0”	Semantic error
length = length(word)	length is not a built-in function name	Syntax error
while length >= longest:	While statement causes infinite loop	Runtime error / semantic error
return longest	This statement is moved to the body of the loop	Syntax error
print longestword(['a', 'apple', 'pear', 'grape'])	In the function name longestword, w is lower case letter.	Syntax error

3 Fill in Blanks in the Program (18 marks)

(Each answer is worth 2 marks)

a) (1) import math (2) 2 (3) break (4) k

b) (1) list, n (2) k = i (3) list[k] < list[j] (4) selsort(list, len(list))

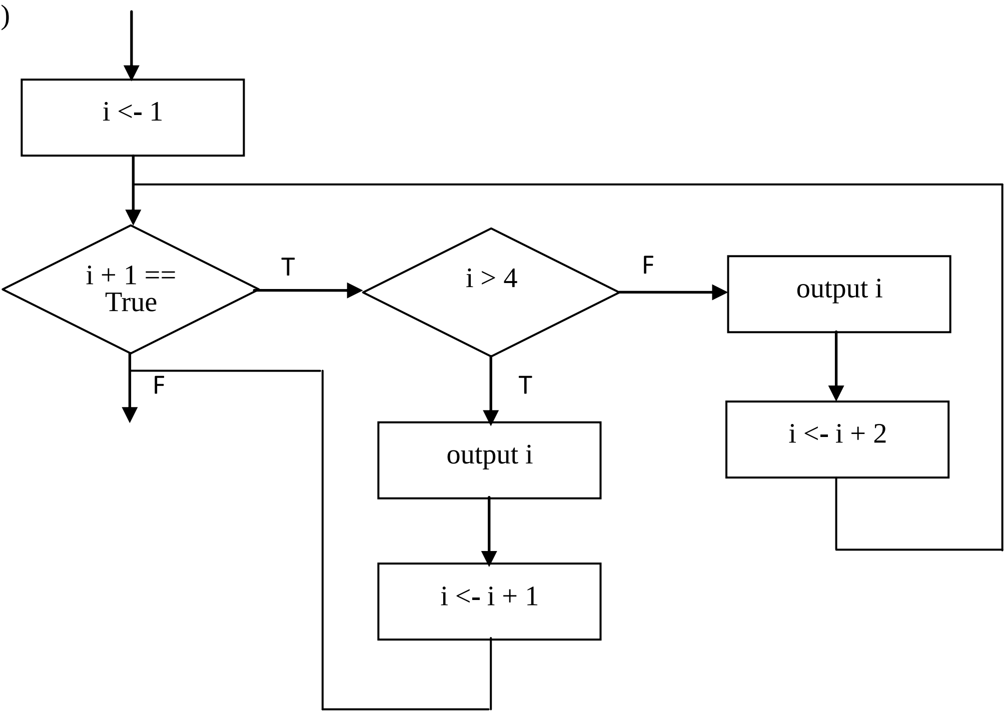
(5) % list[i],

4 Execution of the Program (24 marks)

(Each answer is worth 6 marks.)

a) 3 0 -90 23 6 10 b) raeY weN yppaH

c)



1
3
5

d) True

5 Problem-solving Programming (20 marks)

a) Problem type: search problem or find MCSS problem (3 marks)

b) Solving strategy: (5 marks)

1) Input a sequence at a time, then transform this string into a list and return it.
2) If the current sum of sub-sequence is greater than zero, add current integer to the sum. Otherwise, replace the current sum of sub-sequence with current integer. Record current index as the first index of the sub-sequence.
3) If the current sum of sub-list is greater than previous sum, replace the latter with the former. Record current index as the last index of the sub-sequence.
4) If the integer in the sequence is all checked, return the first and last indexes of the sub-sequence as well as its MCSS.

c) Python program: (12 marks)

```
    def input_sequence():
        seq_str = raw_input("Please input an interger sequence separated by a space, which
                                includes positive, negative or zero:\n")
        seq_list = seq_str.split()
        for i in range(len(seq_list)):
            seq_list[i] = eval(seq_list[i])
        return seq_list

    def findMCSS(n, seq):
        sum = 0
        b = 0
        j = 0
        k = 0
        for i in range(n):
            if b > 0:
                b = b + seq[i]
            else:
                b = seq[i]
                j = i
            if b > sum:
                sum = b
                k = i
        return j, k, sum

    def main():
        seqlist = input_sequence()
        j, k, mcss = findMCSS(len(seqlist), seqlist)
        print "MCSS =",
        for i in range(j, k, 1):
            print seqlist[i], "+",
        print seqlist[k], "= %d" % mcss

main()
```