Section A

Q	A	Explanation
1	A	It is easy to see that C is true whenever B is true or A is false, and C is false otherwise.
2	B	`repeat...until` loop (or `do...while` loop in C) is a post-condition loop, so its loop body will be executed at least once. Therefore at least one '`H`' will be printed.
3	A	Expression (ii) does not work when, for example, `r` is a negative integer. Expression (iii) does not work either because 3.141592654 is not the exact value of π and the result will be affected by precision error.
4	D	A floating point variable cannot be used as an array index.
5	C	Only nine `*`'s will be printed for Choice C.
6	B	`x` get incremented only when `j` equals 5, which happens only when `i` is greater than or equal to 5. By observing the for loop for `i`, we know `x` is incremented only once.
7	A	`char` occupies only 1 byte while other data types occupy more memory.
8	A	This is just a simple dry running problem.
9	B	If there is a syntax error, the program cannot be compiled and no ".exe" file will be generated. To our surprise, many contestants did not choose the correct answer for this problem. Perhaps many contestants are unclear about the meanings of different errors.
10	C	The `while` loop effectively computes the largest power of 2 which divides `N`. Since 2004 is divisible by 4 but not by 8, the answer is 4.
11	B	In the question, we may swap any two numbers, even if they are not next to each other. Since four numbers in the given list are misplaced, at most three swaps are needed, because we may swap twice to put any two of the misplaced numbers in their correct places, then swap once more to make the list sorted in ascending order. Since we cannot sort the given list by only two swaps (which happens only when the four misplaced numbers form two pairs), the minimum number of swaps needed is three.
12	D	It may happen that one basket contains all 2004 apples, so both statements may be false.
13	A	The ranges of possible values for the four choices are: -99 to 99 for Choice A, -99 to 100 for Choice B, -99 to 0 for Choice C and -98 to 98 for Choice D.
14	C	A counter example to expression A is A=0, B=1. A counter example to expression B is A=-1, B=0. C is always true because the exponentiation function is strictly increasing for real numbers.
15	A	Divisions in general introduce precision errors. Loosely speaking, less divisions introduce less errors.
16	A	Program segment B does not work when `m` < 0. Program segment C does not work when `m` < 0 or `n` < 0.
17	B	Program segment A obviously works. Program segment B also works by simple algebraic manipulations. Note that the result remains correct even if `a+b` causes an overflow error. Program segment C does not work when, for instance, `b` equals 0 or when `a*b` causes an overflow error.
18	C	If `m` is a Hamming number and `m` is divisible by 2, then `m`/2 must be a positive integer. Moreover, `m`/2 cannot have prime factors other than 2, 3 or 5. Therefore statement (i) is true. By the same argument, statement (ii) is also true.
19	B	To apply Binary Search to an array, the array must be sorted so as to give the correct result (ascending order and descending order are both fine). Moreover, to determine whether a number exists in an array, the numbers in the array need not be unique.
20	B	To truncate a number to 2 decimal places, the simplest way is to multiply it by 100, truncate the product and then multiply the result by 0.01 (or divide the result by 100).
21	D	It is easy to see that `H` and `K` will be printed when `n<50` and `n>50` respectively. Therefore `O` will be printed when `n=50`, but `I` never gets printed.
22	C	This problem actually asks you to count the number of pairs `(i, j)` such that `i` and `j` are both positive integers less than or equal to 10 and `i` is a multiple of `j`. By listing all pairs systematically, we know there are 27 such pairs.
23	D	This is just a simple dry running problem.
24	B	Note that when the function `f` is called, the parameter `a` is passed by value, so the value of `A` in the main program does not change after calling `f`.

Section B

Q	Possible Answers (Not exhaustive)		Explanation
	Pascal Version:	**C/C++ Version:**	
A	`div 2`	`/2`	The simplest way to reverse a string is to swap the first and the last characters, swap the second and the second last characters, and so on.
B	`len+1-i`	`len-1-i`	
C	`len+1-i`	`len-1-i`	
D	`(n+m-1) div m`	`(n+m-1)/m`	In this question, contestants are asked to compute $\lceil n/m \rceil$, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x (this is called the *ceiling* of x). Many contestants gave answers similar to the following mathematical expression: $\text{round}(n/m+0.49999)$, which is in fact prone to precision error.
E	A=True,B=True,C=False,D=True		This problem can be solved by exhausting all 16 possibilities.
F	`(i=j) or (i=8-j)`	`i==j || i==8-j`	All ‘`X`’s lie on the two diagonals, therefore an ‘`X`’ should be printed if and only if `i` equals `j` or `i+j` equals 8.
G	`n div 3`	`n/3`	A number n is a power of 3 if and only if its base 3 representation contains a single ‘`1`’ and all their other digits are ‘`0`’s. Therefore we may first keep dividing the number by three until it is not divisible by three, effectively removing all trailing ‘`0`’s, then check whether the remaining digits comprise a single ‘`1`’ only.
H	`n=1`	`n==1`	
I	89		This problem can be solved easily if one computes `f(1)`, `f(2)`, ..., `f(10)` in order.
J	n-1		By comparing two distinct integers, one number is eliminated from the list of candidates for the smallest integer. Therefore starting with n candidates, at least $n-1$ comparisons are needed to eliminate $n-1$ candidates. It is simple to see that $n-1$ comparisons suffices to pick out the smallest integer.
K	`22-3*i`	`22-3*i`	This problem is simple by observing that each number (except the first) in the sequence is 3 less than the previous one.
L	n(m+2)		We may solve this problem by observing how `x` varies with `n`, keeping `m` constant, and how `x` varies with `m`, keeping `n` constant.
M	`trunc(sqrt(n))`	`sqrt(n)`	This problem is straight-forward if a contestant understands the prime finding algorithm well.
N	`n mod i = 0`	`n % i == 0`	
O	`s[len]`	`s[len-1]`	Rotating a string can be done by storing the first character in the string in a temporary variable `t`, copying each character left, and finally putting the content of `t` to the last character in the string.
P	`1`	`0`	