CyberChallenge.IT 2019 - Pretest
Commented solutions

Contents

1 Question 1

1.1 Question

Given the following code:

```
char* str = (char*)malloc(4*sizeof(char));
strcpy(str, "...");
```

What is the maximum number of characters that can be put between quotes to avoid a possible memory corruption?

1.2 Answers

(A) 3

(B) 4

(C) 5

(D) The malloc call is invalid

1.3 Proposed solution

The correct answer is `(A) 3`.

The string `str` has room for 4 characters, but strings in C are null terminated, so it is possible to write up to 3 characters (excluding the null byte) without writing in memory out of the bounds of `str`.

2 Question 2

2.1 Question

Given the following array:

```
#define N 3
int a[N][N] = {1,2,3,4,5,6,7,8,9};
```

Which of the following expressions accesses the element of value `5`, i.e., `a[1][1]`?

2.2 Answers

(A) `*a(1*N+1)`

(B) `**a(1*N+1)`

(C) `**(a+1*N+1)`

(D) `*(*(a+1)+1)`

2.3 Proposed solution

The correct answer is `(D) *(*(a+1)+1)`.

Given a pointer `p`, the expressions `p[n]` and `*(p+n)` are equivalent. Therefore `a[1][1]` is the same as `*(a+1)[1]`, which is the same as `*(*(a+1)+1)`.

3 Question 3

3.1 Question

The following (incomplete) implementation of the `strcat` standard function should concatenate the C string `s2` to `s1`:

```
char* strcat(char* s1, const char* s2) {
    char* s = s1;
    while(*s1) s1++;
    do {
        *s1++ = *s2;
    } while(???);
    return s;
}
```

What would you put in place of `???` to make it work?

3.2 Answers

(A) *++s2

(B) *s2++

(C) *s1

(D) The implementation is wrong whatever we put in place of `???`

3.3 Proposed solution

The correct answer is `(B) *s2++`.

Since strings in C are null terminated, `strcat` has to copy the string pointed by `s2` up to its null byte (included). Therefore, the while loop should stop when `s2` points to the null byte. Among the answers, we have:

- `*++s2`, which increments `s2` before dereferencing, and thus prevents the null byte from being copied;
- `*s2++`, which dereferences `s2` before incrementing it, thus the while loop stops just after the null byte is copied;
- `*s1`, which does not increment `s2` but instead works on `s1`, so the first byte of the second string is copied until the first null byte in memory is found (starting from the byte pointed by `s1`).

4 Question 4

4.1 Question

Consider the following function:

```
void foo() {
    void* bar = malloc(320);
    void* baz = &((long*)bar)[???];
    printf("%ld\n", (char*)baz - (char*)bar);
    free(bar);
}
```

Assuming that `sizeof(long)==4`, what would you put in place of `???` so that the function prints 20?

4.2 Answers

(A) 4

(B) 5

(C) 8

(D) Nothing (the function does not compile).

4.3 Proposed solution

The correct answer is `(B) 5`.

The function prints how many chars there are between `baz` and `bar`[1], so, if we substitute `???` with a value `x`, the value printed will be `x*sizeof(long)/sizeof(char)`, which is `x*4`. Thus, in order to obtain 20, `x` should be equal to 5.

[1] See `https://cplusplus.com/doc/tutorial/pointers/#arithmetics`; it is a tutorial for C++, but this section is valid also for C

5 Question 5

5.1 Question

Consider the following C function:

```
void f() {
    struct {
        char s1[3];
        char s2[5];
    } s;

    strcpy(s.s1, "ok");
    strcpy(s.s2, "flag");

    printf("%s\n", ???);
}
```

What would you put in place of `???` so that the function prints `flag`?

5.2 Answers

(A) `&((char*)s)[3]`

(B) `((char*)&s)[3]`

(C) `&((char*)&s)[3]`

(D) The compiler generates an error whatever we put in place of `???`

5.3 Proposed solution

The correct answer is `(C) &((char*)&s)[3]`.

In C, two consecutive fields of a structure are also consecutive in memory. This means that, if we look at `s` as an array of char, it looks like {`'o', 'k', '\0', 'f', 'l', 'a', 'g', '\0'`}.

The expression `&((char*)&s)` is needed to treat the structure as an array of characters, while the index `[3]` is needed to skip the first string of the structure.

6 Question 6

6.1 Question

Consider the following function:

```
void f(char* s) {
    do *s = *s > 'a' && *s < 'z' ? *s - 'a' + 'A' : *s;
    while (*s++);
}
```

What does the call `f(b)` do to a buffer `b` initialized with the string `"guess"`?

6.2 Answers

(A) It leaves `b` unchanged

(B) It converts all letters of `b` to uppercase

(C) It converts just some letters of `b` to uppercase

(D) It generates a runtime error

6.3 Proposed solution

The correct answer is `(B) It converts all letters of b to uppercase`.

The given function uses a ternary operator that is equivalent to the following:

```
void f(char* s) {
    do{
        if(*s > 'a' && *s < 'z'){
           *s = *s - 'a' + 'A';
        }
        else{
            *s = *s;
        }
    while (*s++);
}
```

Therefore, `f` converts all the letters in the range `b-y` to uppercase (the letters `a` and `z` are excluded since there is a strict comparison) and, since every letter of the string `"guess"` is in that range, the entire string is converted.

7 Question 7

7.1 Question

Consider the following C function:

```
int* f(int x) {
    unsigned char i = (unsigned char)x;
    int* v = (int*)malloc(100*sizeof(int));
    if (v != NULL && (char)i < 100) v[i] = x;
    return v;
}
```

Choose a value for `???` in the call `f(???)` so that the array `v` will be written outside of its boundaries. Assume that `sizeof(char)==1`, i.e., 8 bits wide.

7.2 Answers

(A) 110

(B) 250

(C) 256

(D) It is not possible to write outside of the array's boundaries

7.3 Proposed solution

The correct answer is `(B) 250`.

The problem of this function is that, inside the `if` statement, it casts `i` from an unsigned char to a char. This means that values in the range `[128, 255]` are interpreted as values in the range `[-128, -1]`[2], but the following assignment treats them as unsigned again.

This implies that 250 is interpreted as -6, which is obviously less than 100, and it allows to write in position 250, which is outside the boundaries of `v`.

Instead, 256 would not work since `x` is casted to an unsigned char and thus the value 256 would be mapped to 0.

[2] See `https://en.wikipedia.org/wiki/Two's_complement`

8 Question 8

8.1 Question

"If Alice does not eat pizza, then Bob does not drink beer". If this statement is true, which of the following is certainly true?

8.2 Answers

(A) Alice and Bob cannot eat pizza together

(B) If Bob does not drink beer, then Alice eats pizza

(C) If Bob does not drink beer, then Alice does not eat pizza

(D) If Bob drinks beer, then Alice eats pizza

(E) If Alice eats pizza, then Bob drinks beer

8.3 Proposed solution

The correct answer is `(D) If Bob drinks beer, then Alice eats pizza`.

Let us denote by p the proposition "Alice does not eat pizza" and by q the proposition "Bob does not drink beer". Then, the statement in the question is represented by $p \Rightarrow q$ and, by the transposition rule[3], if $p \Rightarrow q$ is true, then also $\overline{q} \Rightarrow \overline{p}$ is true, which corresponds to the statement "If Bob drinks beer, then Alice eats pizza".

[3] See `https://en.wikipedia.org/wiki/Transposition_(logic)`

9 Question 9

9.1 Question

There are three types of security experts: Cyber Defenders always say the truth, Crackers always lie, and Lamers may both say the truth or lie.
Let us suppose that in a group of four people each one says a sentence:

- Alice says: "Bob is a Cyber Defender"
- Bob says: "Charlie is a Lamer"
- Charlie says: "Mallory is a Lamer"
- Mallory says: "Alice is a Cracker"

What is the maximum number of Cyber Defenders that you could find in this group?

9.2 Answers

(A) There are no Cyber Defenders

(B) At most 1 Cyber Defender

(C) At most 2 Cyber Defenders

(D) At most 3 Cyber Defenders

(E) At most 4 Cyber Defenders

9.3 Proposed solution

The correct answer is `(C) At most 2 Cyber Defenders`.

Let us divide the solution in cases:

Alice is a Cyber Defender. Then, also Bob is a Cyber Defender, but Charlie is a Lamer and Mallory is either a Lamer or a Cracker, since she lies. In total there are 2 Cyber Defenders.

Bob is a Cyber Defender. If Alice is a Cyber Defender the case is equivalent to the previous one and there are 2 Cyber Defenders.
If Alice is not a Cyber Defender she must be a Lamer since is telling the truth. Charlie is a Lamer since Bob is telling the truth and Mallory must be a Cracker or a Lamer as in the previous case. In total there is 1 Cyber Defender

Charlie is a Cyber Defender. Then, Mallory is a Lamer, but Alice and Bob cannot be Cyber Defenders, since that implies that Charlie is a Lamer, which is a contraddiction. In total there is 1 Cyber Defender.

Mallory is a Cyber Defender. Then Alice is a Cracker and Bob is not a Cyber Defender. Since Charlie lies, the only Cyber Defender is Mallory. In total there is 1 Cyber Defender.

Hence, the maximum number of Cyber Defenders is 2.

10 Question 10

10.1 Question

A herd of sharks is made of 9 sharks ordered by their size. The smaller the shark the faster it swims. Sharks eat whatever they can catch, including other, smaller sharks. However, when a shark attacks a prey it slows down a little bit. Hence the next, faster shark can attack it (bigger sharks are too slow). For instance, if the smaller shark (number 9) attacks a prey, shark number 8 (only) can decide to attack it and so on. A small, fast fish that only the smaller shark can catch passes next to the herd. Should the shark attack?

10.2 Answers

(A) no, it will be eaten for sure by shark 8

(B) no, the risk of being eaten is too high

(C) yes, shark number 8 won't eat

(D) None of them

10.3 Proposed solution

The correct answer is `(C) yes, shark number 8 won't eat`.

In fact, if shark 8 ate shark 9, then shark 7 could eat shark 8. After that, shark 6 could therefore eat shark 7 and so on, until shark 1 eats shark 2. Thus, shark number 9 can eat the fish without any problem.

11 Question 11

11.1 Question

In the woods there are three types of animals that can speak: foxes, snakes, and turtles. The first ones lie only on rainy days, the second ones always lie, the third ones always tell the truth. One day, an explorer talks with four animals. Their statements, reported in the order in which they were said, are:

(A) "Today it's raining";

(B) "The animal that spoke before me lies";

(C) "Today is sunny";

(D) "The animal, which spoke before me, lies or I am a fox".

How many turtles did the explorer speak to at most?

11.2 Answers

(A) 1

(B) 2

(C) 3

(D) 4

(E) It is not possible to determine it

11.3 Proposed solution

The correct answer is `(B) 2`.

Let us consider the two cases when it is raining and when it is sunny.

It is raining. Then, A is a turtle, since it tells the truth. B lies, so it is not a turtle, and so does C. D tells the truth since C lies, so it is a turtle. In total there are 2 turtles.

It is sunny. A lies, so it is a snake, so B tells the truth. In this case, B can be either a turtle or a fox. The same applies to C. Since C does not lie, D cannot be a turtle. In total there are at most 2 turtles.

The maximum number of turtles is 2.

12 Question 12

12.1 Question

A flight from Milan to New York costs $700; from Milan to Frankfurt costs $100; from Milan to London $200; from New York to Boston $100; from Frankfurt to Boston $600; from Frankfurt to New York $400; from Frankfurt to London $150; from London to Boston $450. What is the minimum cost for flying from Milan to Boston, regardless of the number of flights to take?

12.2 Answers

(A) $500

(B) $600

(C) $650

(D) $800

12.3 Proposed solution

The correct answer is `(B) $600`.

From Milan, you can reach:

- Frankfurt, directly with $100;
- London, directly with $200 (or $250 passing through Frankfurt);
- New York, passing through Frankfurt with $500 (or $700 directly);
- Boston, passing through New York with $600 (or $700 passing through Frankfurt and $800 passing through London).

13 Question 13

13.1 Question

Bar is 10 years old; Baz is three times older than Foo, who is younger than Bar. The sum of the age of Bar and Baz is 31.
How old is Foo?

13.2 Answers

(A) 7

(B) 8

(C) 9

(D) 10

13.3 Proposed solution

The correct answer is `(A) 7`.

Since Bar is 10 years old and the sum of the ages of Bar and Baz is 31, Baz is $31 - 10 = 21$ years old. But Baz is three times older than Foo, so Foo is $\frac{21}{3} = 7$ years old.

14 Question 14

14.1 Question

In the magic land of MalwareLand, there are only three types of malware samples: Foo, Boo, and Bar. Each type of malware sample can either run on Desktop or Mobile. The fraction of samples living in MalwareLand is the same for each type. Assuming that 1/3 of malware samples run on Windows (Desktop), 1/3 on Linux (Desktop), 1/3 on Android (Mobile), what is the probability to find a sample Boo that runs on a Desktop system?

14.2 Answers

(A) 1/3

(B) 2/3

(C) 1/9

(D) 2/9

14.3 Proposed solution

The correct answer is `(D) 2/9`.

For the sake of simplicity, we can assume that the types of malware are uniformly distributed among the platforms; this way, we can easily compute the requested probability, because the Desktop platforms are $\frac{2}{3}$ of the total, and the samples Boo are $\frac{1}{3}$ for each platform, thus giving the result of $\frac{2}{3} \cdot \frac{1}{3} = \frac{2}{9}$.