CyberChallenge.IT 2017 - Test
Commented solutions

Contents

1 Question 1

1.1 Question

Consider the following incomplete code fragment. It declares the type of the nodes of a linked list and defines a function del_key such that del_key(&l,k) removes all the nodes that contain key k from the list pointed to by l:

```
typedef struct node_t node_t;
struct node_t {
    int     key;
    node_t* next;
};

void del_key(node_t ** lp, int key) {
    while (*lp != NULL) {
        if ((*lp)->key == key) {
            node_t* dead = *lp;
            *lp = (*lp)->next;
            free(dead);
        }
        ???
    }
}
```

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

1.2 Answers

(A) `if (*lp == NULL) break;`

(B) `else *lp = (*lp)->next;`

(C) `else lp = &(*lp)->next;`

(D) `*lp = (*lp)->next->next;`

1.3 Proposed solution

The correct answer is `(C) else lp = &(*lp)->next;`.

The code inside the fragment should loop over the nodes of the linked list; when the current key matches k, the current node is deleted and pointer to the next node is replaced with the correct value. The missing part is moving the pointer to the next node if the key does not match.

The choice is between (B) and (C), but while the former replaces the pointer (thus skipping the current node), the latter simply changes the reference with the address of the next pointer, which is the correct operation.

2 Question 2

2.1 Question

A possible way to speed up a function that checks whether two arrays of char contain the same values is to unroll the loop and perform the comparisons by looking at the data as if they were of a wider primitive type, hence requiring fewer loop iterations. What would you put in place of `???` in the function below to achieve this goal?

```
typedef long T;
int test(char * a, char * b, unsigned n) {
   int i, step = sizeof(T)/sizeof(char);
   for (i=0; i+step-1 < n; i += step)
       if (???) return 0;
   for (; i<n; ++i)
       if (a[i] != b[i]) return 0;
   return 1;
}
```

2.2 Answers

(A) `(T)a[i] != (T)b[i]`

(B) `((T*)a)[i] != ((T*)b)[i]`

(C) `(T)(a+i) != (T)(b+i)`

(D) `*(T*)(a+i) != *(T*)(b+i)`

2.3 Proposed solution

The correct answer is `(D) *(T*)(a+i) != *(T*)(b+i)`.

The condition we are looking for should determine if the elements of two arrays are different. The index is incremented each by `step`, so the indexing of the arrays is performed before a possible cast. This excludes answer `(B)`.

Answer `(C)` does not have sense if we want to compare the elements of the arrays, since it is only taking into account the values of the pointers, i.e. just some memory addresses.

Finally, we can exclude answer `(A)`, since it is casting the value of a char to a long (e.g. the char `'A'` is casted to 65). Answer `(D)`, instead, accesses `a` and `b` as char arrays, then it casts them to long arrays and picks the first value of them, which is what is needed.

3 Question 3

3.1 Question

Consider the following C authentication fragment:

```
unsigned password = ??, b = 35, res = password ^ (b & 0xF7);
if (res==0x3A) printf("access granted\n");
```

What value should ?? be so that the program prints “access granted”?

3.2 Answers

(A) 8

(B) 13

(C) 25

(D) 61

3.3 Proposed solution

The correct answer is `(C) 25`.

We have the value of `b`, so we can compute `b & 0xF7 = 35`.

Let us now recall that the XOR is the inverse of itself, so, if `res = password ^ (b & 0xF7)`, then `password = res ^ (b & 0xF7)`. We want `res` to be `0x3A`, so the value of password must be `0x3A ^ 35 = 25`.

4 Question 4

4.1 Question

Obfuscation is a common technique in computer security and software engineering to conceal the semantics of a program for protecting intellectual property or making the code less vulnerable to hacker attacks. The function below computes a function much simpler than it appears:

```
char slp_fitbit(char x) {
    const static unsigned char _[2][2][2][2] = {
        { { { 0xF, 0xE }, { 0xD, 0xC } },
          { { 0xB, 0xA }, { 0x9, 0x8 } } },
        { { { 0x7, 0x6 }, { 0x5, 0x4 } },
          { { 0x3, 0x2 }, { 0x1, 0x0 } } }
    };
    return _[!(x & 8)][!(x & 4)][!(x & 2)][!(x & 1)] | (x & 240);
}
```

4.2 Answers

(A) `slp_fitbit(x) = x`

(B) `slp_fitbit(x) = x % 128`

(C) `slp_fitbit(x) = -x`

(D) `slp_fitbit(x) = x & 0xF0`

4.3 Proposed solution

The correct answer is `(A) slp_fitbit(x) = x`.

The multi-dimensional array is accessed looking at particular bits of the input `x`:

- the first index is 0 if the fourth bit (starting from the right) is 1 and viceversa. All the elements of the first half of the array have in common the fourth bit set to 1, while the elements of the second half have the fourth bit set to 0. This means that the fourth bit of the elements is 1 if and only if the fourth bit of `x` is 1;
- the second index is 0 if the third bit (starting from the right) is 1 and viceversa. In each half of the array, there is a structure analogue to the previous one (the third bit is 1 in the first half and 0 in the second one). As before, the third bit of the elements is 1 if and only if the third bit of `x` is 1;
- the situation is similar for the other two indexes.

The conclusion is that `_[!(x & 8)][!(x & 4)][!(x & 2)][!(x & 1)]` has the four least significant bits equal to the bits of `x` in the same positions; then, a bitwise OR with `x & 240` is performed, and since the binary expression of 240 is `11110000`, this means that we are doing a "concatenation" between the four highest bits and the four lowest bits of `x`.

Since `x` is a char, which is 8 bits, we can deduce that the entire function is simply returning `x`.

5 Question 5

5.1 Question

You are given this simple dynamically-sized array implementation:

```c
#include <stdlib.h>
#include <assert.h>

#define INITIAL_CAPACITY (1<<3)
#define MAX_CAPACITY     (1<<10)

typedef struct {
    int *  data;
    size_t used;
    size_t capacity;
} darray;

darray * init() {
    darray * da = malloc(sizeof(darray));
    assert(da != NULL);
    da->capacity = INITIAL_CAPACITY;
    da->data = malloc(sizeof(int) * da->capacity);
    assert(da->data != NULL);
    da->used = 0;
    return da;
}

void grow(darray * da) {
    da->capacity *= 2;
        if (da->capacity > MAX_CAPACITY)
    exit(EXIT_FAILURE);
    da->data = realloc(da->data, da->capacity);
    assert(da->data != NULL);
}

void add_element(darray * da, int x) {
    assert(da != NULL && da->data != NULL);
    if (da->used == da->capacity)
        grow(da);
    da->data[da->used] = x;
    da->used += 1;
}

int main() {
    darray * da = init();
    int k;
    for (k = 0; k < 512; k++)
        add_element(da, k * 2);
    free(da->data);
    free(da);
    return 0;
}
```

Unfortunately, there's a serious bug: in which function?

5.2 Answers

(A) `init()`

(B) `grow()`

(C) `add_element()`

(D) `main()`

5.3 Proposed solution

The correct answer is `(B) grow()`.

The problem in this code is at the line
`da->data = realloc(da->data, da->capacity);`
inside the function `grow`. Let us recall its syntax:
`void *realloc(void *ptr, size_t size)`
where `size` is the new size for the memory block, in bytes.

Since `da->capacity` is just the number of elements of `data`, we need to multiply that value by the size in bytes of one element. The correct instruction is therefore:
`da->data = realloc(da->data, sizeof(int) * da->capacity);`.

6 Question 6

6.1 Question

Consider the following code fragment:

```
unsigned BB[] = ???;
printf("%s\n", (char *) (BB + 1));
```

Assumptions:

- 32-bit little-endian platform
- sizeof(char) == 1
- sizeof(unsigned) == 4

How would you initialize (i.e., replace `???` above) BB so as to get the output "Say my name"?
Just recall that the decimal ASCII code of the space is 32.

6.2 Answers

(A) {`0xA5207961, 0x536E2079, 0x6D656D61, 0x0`}

(B) {`0xA5, 0x53617920, 0x6D79206E, 0x616D6500`}

(C) {`0xA5536179, 0x206D7920, 0x6E616D65, 0x0`}

(D) {`0xA5, 0x20796153, 0x6E20796D, 0x656D61`}

6.3 Proposed solution

The correct answer is `(D)` {`0xA5, 0x20796153, 0x6E20796D, 0x656D61`}.

The first thing to notice is that `(BB + 1)` performs an increment of the value of the pointer `BB`, so the first element of the array of unsigned is skipped. The string would then start from the second element of `BB`.

Then, the array is accessed as if it was an array of char, but since the platform is little-endian, the first char read is the least significant byte of the corresponding unsigned.

It is now possible to conclude, even without looking at an ASCII table. It is specified that the code for the space is 32 (or `0x20` in hex) and only in one of the proposed answers the first space is the fourth character.

7 Question 7

7.1 Question

In his last trip around Central Italy, Barry filled the gas tank to the top, a total of 40 liters. He traveled at 60 km/h across secondary roads and he knew that his car could make on average 10 km per liter. However, the moment he started, the gas tank developed a leak and 4 hours later the car stopped having run out of gas from the hole.
How many liters of gas had it lost through the leak?

7.2 Answers

(A) 12

(B) 14

(C) 16

(D) 18

7.3 Proposed solution

The correct answer is `(C) 16`.

Going at 60km/h for 4 hours, Barry traveled 240 km; his car consumes a liter every 10 km, so the total consumption is of $\frac{240}{10} = 24$ liters. Because of the leak, Barry lost $40 - 24 = 16$ liters of gas.

8 Question 8

8.1 Question

A deck of French playing cards includes thirteen ranks (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K) for each of the four French suits (clubs, diamonds, hearts, spades).
What is the minimum number of cards you must take to be sure to pick at least one four-of-a-kind (i.e., four cards of the same rank)?

8.2 Answers

(A) 39

(B) 40

(C) 44

(D) 49

8.3 Proposed solution

The correct answer is `(B) 40`.

The worst case is when we take 3 cards of the same rank, for each rank, before we manage to get a four-of-a-kind. There are 13 ranks, so this means that we must pick at most $3 * 13 + 1 = 40$ cards in order to be sure that we have four cards of the same rank.

9 Question 9

9.1 Question

You have 100 kg of watermelons, and 99 percent of their weight comes from water. You let them dehydrate until they are 98 percent made of water.
How many kilograms do they weigh now?

9.2 Answers

(A) 50

(B) 98.09

(C) 98.9

(D) 98.99

9.3 Proposed solution

The correct answer is `(A) 50`.

At the beginning, the weight of the watermelons is divided into 99 kg of water and 1 kg of pulp, dehydrating the watermelons does not affect the pulp, so at the end we still have 1 kg of pulp, which should represent the $100 - 98 = 2$ percent of the total weight. This means that the total weight is $\frac{1}{\frac{2}{100}} = 50$ kilograms.

10 Question 10

10.1 Question

Oliver, Sara, Shado, and Slade are all trapped in Lian Yu, an island in the middle of a crocodile infested lake. They have one crocodile repelling stick that protects only up to two swimmers. To get to safety a maximum of two swimmers can be in the water at the same time, also they have to be together to benefit from the stick and they have to swim at the pace of the slower swimmer.
As the shore is too far, someone has to swim back with the stick until all four are safe on the shore. Oliver can swim the distance in 3 minutes, Sara in 7, Shado in 13, and Slade in 17.
What is the minimum time required for them to all get to safety?

10.2 Answers

(A) 41

(B) 43

(C) 46

(D) 47

10.3 Proposed solution

The correct answer is `(A) 41`.

Let us notice that if Shado and Slade do not swim together, there will be a lot of wasted time. We are therefore looking for a strategy in which they swim together. The strategy is the following:

- Oliver and Sara swim to the shore (7 minutes)
- Oliver swims back to the island (3 minutes)
- Shado and Slade swim to the shore (17 minutes)
- Sara swim back to the island (7 minutes)
- Oliver and Sara swim to the shore (7 minutes)

It took $7 + 3 + 17 + 7 + 7 = 41$ minutes to them to get to safety.

11 Question 11

11.1 Question

Four good friends visited Bruce at his manor last year for his birthday. His butler Alfred took notes on what time each of them arrived, but he forgot to write down whether it was before (AM) or after (PM) noon. According to his notes:

- James arrived at 8:00
- Lee arrived at 9:00
- Selina arrived at 10:00
- Lucius arrived at 11:00

While Bruce recalls that:

- Selina did not visit him between Lee and Lucius
- At least one friend visited him between James and Lee
- James might have visited him before Selina or Lucius, but not before both

Which of the following options is compatible with the events as described above? (hint: there is only one feasible assignment of the AM/PM suffix to the events).

11.2 Answers

(A) `James 8:00 AM`

(B) `Lee 9:00 PM`

(C) `Selina 10:00 AM`

(D) `Lucius 11:00 AM`

11.3 Proposed solution

The correct answer is `(D) Lucius 11:00 AM`.

If James did not visit Bruce before both Selina and Lucius, his visit must have been at 8:00 PM. Otherwise James would be the first to visit Bruce and Selina and both Lucius would have visited him after James.

If someone visited Bruce between James and Lee, Lee's visit must have been at 9:00 AM. Otherwise, there could not be any visit between 8 PM and 9 PM.

If Selina did not visit Bruce between Lee and Lucius, then Lucius must have been the second to visit Bruce (at 11:00 AM) and Selina must have visited Bruce at 10:00 PM.

12 Question 12

12.1 Question

Fobi govv, iye nsn bowowlob Tevsec Mkockb. Grkd sc dro cew yp kvv dro xewlobc pbyw yxo dy pybdi?
Hint: we used a substitution cipher that replaces each letter of the alphabet with another.

12.2 Answers

(A) 780

(B) 800

(C) 820

(D) 840

12.3 Proposed solution

The correct answer is `(C) 820`.

It is reasonable to assume that the message is in English and, even though the given string is too short, we can try to perform some kind of statistical analysis: the most frequent character is 'o', while in English it is 'e', so we can assume that 'o' is mapped to 'e' while decoding the string.

Moreover, we can assume (or at least try with this assumption in the beginning) that the cipher is similar to the Caesar Cipher, in which every character is mapped to the character in the alphabet at a fixed "distance" (modulo 26). This way, we would have 'o' → 'e', 'p' → 'f', and so on, where the distance is 16.

The decoded message is
`Very well, you did remember Julius Caesar.  What is the sum of all the numbers from one to forty?`.
The answer to the question is $\frac{40 \cdot 41}{2} = 820$.