CS100 Fall 2023

Quiz 1

Oct 26, 2023

Answer the questions according to the C17 standard.

1. (15 points) Your name: ______________. Your student ID: ______________.
   Your email: ______________`@shanghaitech.edu.cn`

2. (15 points) Select the piece(s) of code that have undefined behaviors.

   A.
   ```
   int random(void) { int x; return x; }
   int main(void) { printf("The random number is %d\n", random()); }
   ```

   B.
   ```
   int table[10];
   int exists_in_table(int value) {
     for (int i = 0; i <= 10; ++i)
       if (table[i] == value) return 1;
     return 0;
   }
   ```

   C.
   ```
   double *ptr = NULL; free(ptr);
   ```

   D.
   ```
   void func(int, int);
   int main(void) { int x = 42; func(x, x + 1); }
   ```

   E.
   ```
   void print_as_int(double x) {
     printf("%d\n", x);
   }
   ```

   F.
   ```
   void foo(int *ptr) { // ptr may be null
     if (*ptr == 42 && ptr) do_something();
   }
   ```

3. (10 points) Design a function `swap` that can swap two pointers (both of type `int *`).

   ```
   int i = 10, j = 15, *p1 = &i, *p2 = &j;
   swap(/* you design this */);
   ```

   After this call to `swap`, `p1` should be pointing to `j` and `p2` should be pointing to `i`.

   A. `void swap(int *p1, int *p2) { int *tmp = p1; p1 = p2; p2 = tmp; }`

      To call this function: `swap(p1, p2);`

   B. `void swap(int **p1, int **p2) { int *tmp = *p1; *p1 = *p2; *p2 = tmp; }`

      To call this function: `swap(&p1, &p2);`

   C. `void swap(int **p1, int **p2) { int **tmp = p1; p1 = p2; p2 = tmp; }`

      To call this function: `swap(&p1, &p2);`

   D. `void swap(int *p1, int *p2) { int **tmp = &p1; &p1 = &p2; &p2 = tmp; }`

      To call this function: `swap(*&p1, *&p2);`

4. (10 points) Suppose `n` is an unsigned integer. Select the piece(s) of code where `foo` is always called exactly `n` times.

   A. `while (n--) foo();`

   B. `for (unsigned i = 0; i <= n; ++i, foo()) {}`

   C.
   ```
   unsigned i = 0;
   do { foo(); ++i; } while (i < n);
   ```

   D.
   ```
   unsigned i = 0;
   while (1) {
     if (++i >= n) break;
     foo();
   }
   ```

5. (10 points) Read the following code. Select the correct statement(s).

```
#include <stdio.h>
int x = 42;                 // (1)
int incre(int x) {          // (2)
  return x + 1;
}
int main(void) {
  int x;                    // (3)
  x = 15;
  if (x == 15) {
    int x = incre(10);
    printf("%d\n", x++);    // (4)
  }
  printf("%d\n", incre(x)); // (5)
  return 0;
}
```

   A. The `x` in (1) and (2) refer to the same variable.

   B. In (3), the value of `x` is `42` because it was initialized to `42` in (1).

   C. The value printed by (4) is `11`.

   D. The value printed by (5) is `16`.

   E. The value printed by (5) is `13`.

6. (20 points) For each piece of code, write down the type of `var`.

   (a) `double* a, b[10], var;`

   (a) `double`

   (b) `void foo(int var[3][4]) { /* ... */ }`

   (b) `int (*)[4]`

   (c) `int var[3][4], *x = &var[0][0];`

   (c) `int [3][4]`

   (d) `char *var[] = {"hello", "world"};`

   (d) `char *[2]`