Common Bugs in C/C++ Programming

Most of the contents are directly from or modified from Prof. Liu Pangfeng's blog (https://pangfengliu.blogspot.com). Most credits should go to him.

For the following problems, answer the output message of the program unless they are specified differently. If there are any exceptions (or segmentation faults), indicate the location in the source code. Most importantly, you need to explain the reason.

Problem 1: Array v.s. Pointer

```
#include <stdio.h>
#include <string.h>
int main(void) {
    char *start = "this is a string";
    start[4] = '\0';
    printf("%s\n", start);
}
```

```
#include <stdio.h>
#include <string.h>
int main(void) {
    char start[] = "this is a string";
    start[4] = '\0';
    printf("%s\n", start);
}
```

Problem 2

Show the output and explain the difference of the two loops.

```
#include <stdio.h>
#include <string.h>
int main() {
  char string[] = "this is a string";
  char *start;
  start = string;
  start = strtok(start, " ");
  while (start != NULL) {
    printf("%s\n", start);
    start = strtok(NULL, " ");
  }
  start = string;
  start = strtok(start, " ");
  while (start != NULL) {
    printf("%s\n", start);
    start = strtok(NULL, " ");
  }
}
```

Problem 3

Explain why a character is missing.

```
#include <stdio.h>
int main() {
  FILE *fp = fopen("file", "wb");
  for (int i = 0; i < 256; i++)
    fputc(i, fp);
  fclose(fp);
  fp = fopen("file", "rb");
  int count = 0;
  char c;
  while ((c = fgetc(fp)) != EOF)
    count++;
  printf("count = %d\n", count);
}
```

Problem 4

Why is our lab number incorrect?

```
#include <stdio.h>
int main() {
  long int lab_tel = 035731603;
  printf("my lab’s telephone number is %ld\n", lab_tel);
}
```

Problem 5

```
#include <stdio.h>
int main() {
  int a[10];
  if (a == &a)
    printf("yes\n");
  else
    printf("no\n");

  if (a + 1 == &a + 1)
    printf("yes\n");
  else
    printf("no\n");
}
```

Problem 6

Answer the size of "`file`" in Linux and Windows, and explain.

```
#include <stdio.h>
int main() {
  FILE *fp = fopen("file", "w");
  fputs("hello\n", fp);
  fputs("hello", fp);
  fputs("hello\n", fp);
  fclose(fp);
}
```

Problem 7

Hint: NEVER NEVER run this. Otherwise, your hard disk will crash. Just tell what is wrong with this program.

```
#include <stdio.h>
int main() {
  FILE *fp = fopen("file", "wb");
  for (char c = 0; c < 256; c++) {
    fputc(c, fp);
  }
  fclose(fp);
}
```

Problem 8

```
#include <stdio.h>
#define inc(x) ((x)++)
#define square(x) (x * x)
int main() {
  int i = 3, j = 4;
  printf("%d\n", square(i + j));
  printf("%d %d\n", square(inc(i)), i);
}
```

Problem 9

Hint: this is very very important to the program development in our lab.

```
#include <stdio.h>
struct csie {
  char c;
  short s;
  int i;
  double e;
};
struct ceis {
  char c;
  double e;
  int i;
  short s;
};
int main() {
  printf("csie = %d\n", sizeof(struct csie));
  printf("ceis = %d\n", sizeof(struct ceis));
}
```

Problem 10

```
#include <stdio.h>
#include <string.h>
int main() {
  char source[] = "This is a string.";
  char destination[4];
  int i = 5;
  strcpy(destination, source);
  printf("i is %d\n", i);
  printf("source is [%s]\n", source);
  printf("destination is [%s]\n", destination);
}
```

Problem 11 (The examples are given by Ting-Fu Liao.)

```
// header.h
#include <stdio.h>
static int val = 0;
void set(int x);

// impl.c
#include "header.h"
void set(int x) {
  val = x;
}

// main.c
#include "header.h"
int main() {
  set(100);
  if (val == 100)
    printf("val == 100\n");
  else
    printf("val != 100\n");
}
```

Problem 12

Why can't you open the file?

```
#include <stdio.h>
int main() {
  char filename[80];
  printf("input file name: ");
  fgets(filename, 79, stdin);
  FILE *fp = fopen(filename, "r");
  // try assert(fp != NULL);
  fclose(fp);
}
```

Problem 13

```
#include <stdio.h>
int main() {
  int i = 2147483647;
  unsigned int ui = 2147483647;
  if (i + 1 < 0)
    printf("i + 1 < 0\n");
  if (ui + 1 > 0)
    printf("ui + 1 > 0\n");
  if (ui + 1 > i + 1)
    printf("ui + 1 > i + 1\n");
}
```

Problem 14

```
#include <stdio.h>
int main() {
  unsigned int ui = 2147483647;
  if (ui + 1 > 0)
    printf("ui + 1 > 0\n");
  if (ui + 1 < -1)
    printf("ui + 1 < -1\n");
}
```

Problem 15

```
#include <stdio.h>
int main() {
  int i = -13;
  if ((i / 2) == (i >> 1))
    printf("yes\n");
  else
    printf("no\n");
}
```

Problem 16

```
#include <stdio.h>
#include <stdlib.h>
int compare(const void *a, const void *b) {
  return (*(int *)a - *(int *)b);
}

int main() {
  int values[] = {-2147483640, 50, 100};
  qsort(values, 3, sizeof(int), compare);
  for (int n = 0; n < 3 ; n++)
     printf("%d ", values[n]);
}
```

Problem 17

What is the output? Hint: use "`gcc -E test.c`" to see what happens.

```
#include <stdio.h>
#include <assert.h>
int main() {
  FILE *fp = fopen(__FILE__, "r");
  assert(fp != NULL);
  int c;
  while ((c = fgetc(fp)) != EOF)
    putchar(c);
  fclose(fp);
}
```

Problem 18

```
#include <stdio.h>
#define SWAP(x, y) x ^= y ^= x ^= y
int main() {
  int i = 3, j = 5;
  printf("%d %d\n", i, j);
  SWAP(i, j);
  printf("%d %d\n", i, j);
  SWAP(i, i);
  printf("%d\n", i);
}
```

Problem 19

```
#include <stdio.h>
int main() {
  int i = 3;
  i = i++ + ++i;
  printf("%d\n", i);
}
```

Problem 20

```
#include <stdio.h>
int *bar(int t) {
  int i = t;
  int *temp = &i;
  printf("temp is %d, (*temp) is %d\n", temp, *temp);
  return temp;
}

void foo(int a, int b) {
  int i;
  int *temp = &i;
  *temp = a + b;
}

int main() {
  int *a;
  a = bar(10);
  printf("a is %d, (*a) is %d \n", a, *a);
  foo(10, 20);
  printf("a is %d, (*a) is %d \n", a, *a);
}
```

Problem 21

```
#include <stdio.h>
int main() {
  char i = 1;
  char j;
  scanf("%d", &j);
  if (i & j)
    printf("yes.\n");
  else
    printf("no.\n");
}
```

Input:

```
3
```

Problem 22

Hint: Visual C++ 6.0

```
// 程式將 i 調整為偶數後再乘以 5
#include <stdio.h>
int main() {
  int i = 3;
  // 檢驗 i 是否為奇數
  if (i % 2 == 1) // 成功
```

```
    i++;
  i *= 5; // 變成偶數後再乘以 5
  printf("%d\n", i);
}
```

Problem 23 (The examples are given by Ting-Fu Liao.)

Show the output of the translated program, and run it. Besides, you need to give a scenario when/where you would use it in this way

```
#include <cstdio>
#include <iostream>
#include <map>
#include <string>
#define FuncDef(cmd) void cmd_##cmd() { printf("cmd: "#cmd"\n"); }
#define RegFunc(cmd) m[#cmd] = cmd_##cmd;

std::map<std::string, void(*)()> m;
FuncDef(quit);
FuncDef(help);
int main() {
  RegFunc(quit);
  RegFunc(help);
  std::string cmd;
  while (getline(std::cin, cmd)) {
    if (m.count(cmd)) (*m[cmd])();
    else printf("Not support %s\n", cmd.c_str());
  }
}
```

Problem 24

```
#include <stdio.h>
#include <stdlib.h>
int main() {
  int *p = (int *) malloc(sizeof(int));
  int *q = (int *) realloc(p, sizeof(int));
  *p = 1;
```

```
      *q = 2;
      if (p == q)
        printf("%d %d\n", *p, *q);
    }
```

Problem 25 (The examples are given by Yu-Hsuan Cheng.)

```
    #include <iostream>
    #include <vector>
    #include <numeric>
    #include <functional>
    int main() {
        std::vector<float> v{1.5, 2.5, 3.5};
        float sum = std::accumulate(v.begin(), v.end(), 0);
        std::cout << sum << std::endl;
    }
```

Problem 26 (The examples are given by Yu-Hsuan Cheng.)

```
    #include <iostream>
    #include <string>
    using std::string;
    void add_argument(string name, string long_name, string
desc, bool required = false) {
      std::cout << "long version " << required << std::endl;
    }
    void add_argument(string name, string desc, bool required =
false) {
      std::cout << "short version " << required << std::endl;
    }
    int main() {
      add_argument("-h", "--help", "Show Help Menu");
    }
```

References

- What Every C Programmer Should Know About Undefined Behavior #1/3: http://blog.llvm.org/2011/05/what-every-c-programmer-should-know.html
- A Guide to Undefined Behavior in C and C++, Part 1: https://blog.regehr.org/archives/213
- 萬惡的未定義行為: http://blog.ez2learn.com/2008/09/27/evil-undefined-behavior/