

# zq2 - HW0

---

## Comments

2-didn't print to stderr;2.5: \*(data+3); 3.3: not correct; 4.2: should state more diffs; 4.11:not work for friends; 4.13: should not free age, not a pointer;6.2:partially correct;

## Timestamp

1/22/2018 16:24:20

### 1. Write a program that uses write() to print out "Hi! My name is ".

```
#include<unistd.h>

int main()
{
    write(1, "Hi! My name is Max Qian", 23 );
}
```

Grade: **100.0%**

### 2. Write a program that uses write() to print out a triangle of height n to Standard Error

```
#include <unistd.h>
void write_triangle(int n)
{
    int i;
    for (i = 0; i <= n; i++)
    {
        int j;
        for (j = 0; j < i; j++)
        {
            write(1, "*", 1);
        }
        if(i > 0)
        {
            write(1, "\n", 1);
        }
    }
}

int main(int argc, char *argv[])
{
    if(argc > 1)
    {
        write_triangle(atoi(argv[1]));
    }
    return 0;
}
```

Grade: **50.0%**

**3. Take your program from "Hello World" and have it write to a file called "hello world.txt" (without the quotes).**

```
#include <unistd.h>
void writef()
{
    int filde = open("hello_world.txt", O_CREAT | O_RDWR, S_IRUSR | S_IWUSR | S_IROTH);
    write(filde, "Hi! My name is Max Qian\n", 24);
    close(filde);
}
```

Grade: **100.0%**

**4. Take your program from "Writing to files" and replace it with printf()**

```
void printf()
{
    close(1);
    int filde = open("hello_world.txt", O_CREAT | O_RDWR, S_IRUSR | S_IWUSR | S_IROTH);
    printf("Hi! My name is Max Qian\n");
    close(filde);
}
```

Grade: **100.0%**

**5. Name some differences from write() and printf()**

printf() acts as a buffer which only print when it reaches the maximum capacity or the special mark such as "\n". Also, printf() can take format identifiers. On the other hand, write() requires 3 arguments to execute and it puts the string when the function ends.

Grade: **100.0%**

**1. How many bits are there in a byte?**

8

Grade: **100.0%**

**2. How many bytes is a char?**

1

Grade: **100.0%**

**3. Tell me how many bytes the following are on your machine: int, double, float, long, long long**

4 bytes, 8 bytes, 4 bytes, 4 bytes, 8 bytes

Grade: **100.0%**

4. On a machine with 8 byte integers (refer to code snippet below)

```
0x7fbd9d50
```

Grade: 100.0%

5. What is `data[3]` equivalent to in C?

```
3[data]
```

Grade: 50.0%

6. Why does this segfault (refer to code snippet below)?

```
there is no write permission at the address of ptr
```

Grade: 100.0%

7. What does `sizeof("Hello\0World")` return?

```
12
```

8. What does `strlen("Hello\0World")` return?

```
5
```

9. Give an example of X such that `sizeof(X)` is 3

```
"AB"
```

Grade: 100.0%

10. Give an example of Y such that `sizeof(Y)` might be 4 or 8 depending on the machine.

```
int*
```

Grade: 100.0%

1. Name me two ways to find the length of `argv`

```
1. argc 2. sizeof(argv)
```

Grade: 100.0%

2. What is `argv[0]`

```
the program name
```

Grade: 100.0%

**3. Where are the pointers to environment variables stored?**

they are stored on process's own memory

Grade: 0.0%

**4. On a machine where pointers are 8 bytes (refer to the code snippet)**

8, 6. `sizeof(ptr)` returns the size of `char*`, and `sizeof(array)` returns the size of the string

Grade: 100.0%

**5. What datastructure is managing the lifetime of automatic variables?**

stack

Grade: 100.0%

**1. If I want to use data after the lifetime of the function it was created in, then where should I put it and how do I put it there?**

heap. create the data by using `malloc`

Grade: 100.0%

**2. What are the differences between heap and stack memory?**

stack memory is freed automatically after the function, and heap memory has to be freed manually

Grade: 100.0%

**3. Are there other kinds of memory in a process?**

static

Grade: 100.0%

**4. Fill in the blank. In a good C program: "For every `malloc` there is a \_\_\_\_".**

`free()`

**5. Name one reason `malloc` can fail.**

there is not enough storage for `malloc` to allocate

Grade: 100.0%

## 6. Name some differences between time() and ctime()

ctime returns a string of the time and time returns the seconds since the Epoch  
time

Grade: 100.0%

## 7. What is wrong with this code snippet?

ptr is double freed, the processor would throw exception

Grade: 100.0%

## 8. What is wrong with this code snippet?

ptr has been freed and access this address may read other program's data

Grade: 100.0%

## 9. How can one avoid the previous 2 mistakes?

set the pointer to NULL after free it

Grade: 100.0%

## 10. Create a struct that represents a Person and typedef, so that "struct Person" can be replaced with a single word. A person should contain the following information: name, age, friends (pointer to an array of pointers to People).

```
struct person{
    char* name;
    int age;
    struct person** friends;
}
```

```
typedef struct person person_t;
```

Grade: 100.0%

## 11. Now make two people "Agent Smith" and "Sonny Moore" on the heap who are 128 and 256 years old respectively and are friends with each other.

```
int main(){
    person_t* as = (P_Person*)malloc(sizeof(person_t));
    person_t* sm = (P_Person*)malloc(sizeof(person_t));
    as->name = "Agent Smith";
    as->age = 128;
    as->friends = sm;
    sm->name = "Sonny Moore";
    sm->age = 256;
    sm->friends = as;
}
```

Grade: 50.0%

12. 'create()' should take a name and age. The name should be copied onto the heap. Use malloc to reserve sufficient memory for everyone having up to ten friends. Be sure initialize all fields (why?).

```
person_t* create(char* name, int age)
{
    person_t* new_obj = (person_t*)malloc(sizeof(person_t));
    new_obj->name = name;
    new_obj->age = age;
    new_obj->friends = (person_t*) calloc(sizeof(person_t) * 10);
    return new_obj;
}
```

if we do not initialize all fields, there may be trash values that can mess up the structure.

Grade: 100.0%

13. 'destroy()' should free up not only the memory of the person struct, but also free all of its attributes that are stored on the heap. Destroying one person should not destroy any others.

```
void destroy(person_t* p)
{
    memset(p, 0, sizeof(person_t));
    free(p->name);
    free(p->age);
    free(p->friends);
    free(p);
}
```

Grade: 50.0%

1. What functions can be used for getting characters for stdin and writing them to stdout?

getchar() and putchar()

Grade: 100.0%

2. Name one issue with gets()

if input is too long, there would be an overflow that affects the whole input

Grade: 100.0%

3. Write code that parses a the string "Hello 5 World" and initializes 3 variables to ("Hello", 5, "World") respectively.

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main() {
```

```

        char* ptr = "Hello 5 World";
        char s1[5];
        char s2[6];
        int i;
        sscanf(ptr, "%s %d %s", s1, &i, s2);
        printf("%s, %d, %s\n", s1, i, s2);
    }
    return 0;
}

```

Grade: 100.0%

#### 4. What does one need to define before using getline()?

```
#define _GNU_SOURCE
```

Grade: 100.0%

#### 5. Write a C program to print out the content of a file line by line using getline()

```

#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>

void printLine(FILE* f){
    char* buffer = NULL;
    size_t capacity;
    ssize_t result = getline(&buffer, &capacity, f);
    while((result = getline(&buffer, &capacity, f)) != -1)
    {
        if(buffer[result] == '\n') buffer[result] = 0;
        if(result > 1)printf("%s\n", buffer);
    }
    free(buffer);
    fclose(f);
}

int main(int argc, char *argv[]) {
    FILE* stream = fopen(argv[1], "r");
    if (stream == NULL) {
        perror("fopen");
        exit(EXIT_FAILURE);
    }
    printLine(stream);
    return 0;
}

```

Grade: 100.0%

#### 1. What compiler flag is used to generate a debug build?

```
gcc -g
```

Grade: 100.0%

**2. You modify the Makefile to generate debug builds and type make again. Explain why this is insufficient to generate a new build.**

we have to make clean first in order to update the whole program.

Grade: **50.0%**

**3. Are tabs or spaces used in Makefiles?**

tab

Grade: **100.0%**

**4. What does 'svn commit' do? What's a revision number?**

It pushes the code to the svn repository, and revision number is the identification number of that commit.

Grade: **100.0%**

**5. What does 'svn log' show you?**

It displays commit log messages from the repository.

Grade: **100.0%**

**Final grade: 0.9186046512%**