

1. (20 points) ..... Match the Bug

Match each code snippet with the appropriate bug.

\_\_\_\_\_ use-after-free

\_\_\_\_\_ memory leak

\_\_\_\_\_ segmentation fault

\_\_\_\_\_ null termination bug

\_\_\_\_\_ buffer overflow

---

```
(a) char *s = "Happy Halloween!";
    char buf[strlen(s)];
    strcpy(buf, s);
```

---

```
(b) char *a = malloc(10);
    ...
    a = null;
    free(a);
```

---

```
(c) int *f()
{
    int x = 10;
    return &x;
}
int main()
{
    int* p = f();
    *p = 20;
}
```

---

```
(d) char buf[8];
    char alphanum[] = "abcdefghijklmnopqrstuvwxyz0123456789";
    srand(time(NULL));
    for (int i = 0; i < 8; i++) {
        buf[i] = alphanum[rand() % 36];
    }
    printf("random string: '%s'\n", buf);
```

---

```
(e) char *s = malloc(200);
    s = "BOOOO!";
    free(s);
```

---

## 2. (20 points) ..... Password Cracking

Is the hash `FC1198178C3594BFDDA3CA2996EB65CB` in the following *precomputed hash chain* table of width 5? If so, what is the plaintext? Use the supplied lookup tables. Plaintexts are drawn from the characters and (you may use the P and G characters, respectively, to save time instead of drawing out pumpkins and ghosts). *For full credit, you must show the complete chain containing the answer.*

Answer: \_\_\_\_\_

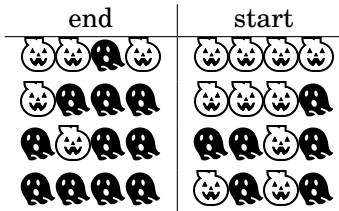
---



---



---



```
func reducer(c):
    Convert the first digit of c using the reduction function table.
```

Hash function lookup table

plaintext	hash of plaintext
	4A7D1ED414474E4033AC29CCB8653D9B
	25BBDCD06C32D477F7FA1C3E4A91B032
	FC1198178C3594BFDDA3CA2996EB65CB
	AE2BAC2E4B4DA805D01B2952D7E35BA4
	DB2F40F24260BC41DB48D82D5E7ABF1D
	814F06AB7F40B2CFF77F2C7BDFFD3415
	2A66ACBC1C39026B5D70457BB71B142B
	7D7C45B9A935CF9D845FC75679A41559
	A9B7BA70783B617E9998DC4DD82EB3C5
	B8C37E33DEFDE51CF91E1E03E51657DA
	1E48C4420B7073BC11916C6C1DE226BB
	7F975A56C761DB6506ECA0B37CE6EC87
	1E6E0A04D20F50967C64DAC2D639A577
	C6BFF625BDB0393992C9D4DB0C6BBE45
	2CBCA44843A864533EC05B321AE1F9D1
	B59C67BF196A4758191E42F76670CEBA

Reduction function

hex digit	plaintext
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	