CSE 363

Final Project: Reverse Engineering

New Mexico Institue of Mining and Technology

Matthew C. Lindeman, Luke Shankin, Nikki Sparacino, Cameron Savage

Date April 17, 2023

1 Heap's Algorithm

```
1  ulong dbg.main(ulong argc, int64_t argv)
2
3  {
4      int32_t iVar1;
5      int64_t arg1;
6      ulong str;
7      ulong var_24h;
8      ulong arr;
9      uint32_t i;
10     int32_t var_8h;
11     ulong val;
12
13     // int main(int argc,char ** argv);
14     val._0_4_ = 7;
15     if (argc == 2) {
16         val._0_4_ = sym.imp.atoi(*(argv + 8));
17     }
18     arg1 = sym.imp.calloc();
19     for (var_8h = 1; var_8h <= val; var_8h = var_8h + 1) {
20         *(var_8h * 4 + -4 + arg1) = var_8h;
21     }
22     iVar1 = dbg.fact(val);
23     for (i = 0; i < iVar1; i = i + 1) {
24         dbg.heaps(arg1, val, i);
25         sym.imp.printf("%d⎾");
26         dbg.printarr(arg1, val);
27         if (i % 6 == 5) {
28             sym.imp.putchar(10);
29         }
30         if (i + ((i / 6 + (i >> 0x1f) >> 2) - (i >> 0x1f)) * -0x18 == 0x17) {
31             sym.imp.puts(0x402009);
32         }
33     }
34     sym.imp.free(arg1);
35     return 0;
36 }
```

Some insightful analysis here.

```
1  void dbg.swap(uint *arg1, uint *arg2)
2
3  {
4      uint uVar1;
5      ulong b;
6      ulong a;
7      ulong tmp;
8
9      // void swap(int * a,int * b);
10     uVar1 = *arg1;
11     *arg1 = *arg2;
12     *arg2 = uVar1;
13     return;
14 }
```

Some insightful analysis here.

```
1  void dbg.printarr(int64_t arg1, ulong arg2)
2
3  {
4      ulong arr;
5      ulong i;
6
7      // void printarr(int * arr,int length);
8      for (i._0_4_ = 0; i < arg2; i._0_4_ = i + 1) {
9          sym.imp.printf(0x40200b, *(arg1 + i * 4));
10     }
11     sym.imp.putchar(10);
12     return;
13 }
```

Some insightful analysis here.

```
1  int32_t dbg.fact(ulong arg1)
2
3  {
4      int32_t iVar1;
5      ulong x;
6
7      // int fact(int x);
8      if (arg1 < 1) {
9          iVar1 = 1;
10     }
11     else {
12         iVar1 = dbg.fact(arg1 - 1);
13         iVar1 = iVar1 * arg1;
14     }
15     return iVar1;
16 }
```

Some insightful analysis here.