Pico Gym Exclusive
Picker IV

```
int win() {
  FILE *fptr;
  char c;

  printf("You won!\n");
  // Open file
  fptr = fopen("flag.txt", "r");
```

This challenge is an example of a ret2win binary hacking challenge

Ret2Win Challenges

```
└─$ ./picker-IV
Enter the address in hex to jump to, excluding '0x':
```

In Ret2Win challenges, the goal of the challenge is to force the program to run a function which it normally wouldn't run under normal circumstances

Ret2Win Challenges

```
└─$ ./picker-IV
Enter the address in hex to jump to, excluding '0x':
```

```
int win() {
  FILE *fptr;
  char c;
```

This program gives us an easy way to run code according to its memory address, so if we know the address of the `win` function, we can run it and complete the challenge

Finding the Function Address

```
└─$ objdump -d picker-IV | grep win
000000000040129e <win>:
  4012d2:       75 16
```

```
[0x7f21f25ad440]> afl | grep win
0x0040129e    6    150 sym.win
```

There are a few different ways to find the memory address of a program function, such as the `objdump` command and the `Rardare2` debugger

Finding the Function Address

```
└─$ objdump -d picker-IV | grep win
000000000040129e <win>:
  4012d2:       75 16
```

```
[0×7f21f25ad440]> afl | grep win
0×0040129e    6    150 sym.win
```

Keep in mind that memory addresses are often expressed in hexadecimal numbers