Value	Representation (bits)	Representation hex	Representation decimal	Useful C type
	31-bit, 2's complement number			
9	0000 0000 0000 0000 0000 0000 0001 **0011**	0x00000013	19	int
-2	1111 1111 1111 1111 1111 1111 1111 **1101**	0xFFFFFFFD	-3	int
true	0000 0000 0000 0000 0000 0000 0000 **0110**	0x00000006	6	int
false	0000 0000 0000 0000 0000 0000 0000 **0010**	0x00000002	2	int
(pair 1 2)	XXXX XXXX XXXX XXXX XXXX XXXX XXXX XX**00**	0xXXXXXXXX	big num	_____

tag bits (X1=num, 10=bool, 00=pair)

```
extern int our_code_starts_here()
  asm("our_code_starts_here");

void print_val(int val) {
  if(val & 1) { printf("%d", (val - 1) / 2); }
  else if (val == 6) { printf("true"); }
  else if (val == 2) { printf("false"); }
  else {









  }
}

int main(int argc, char** argv) {
  int input = 0;

  int* MEMORY = calloc(10000, sizeof(int));

  if(argc > 1) { input = atoi(argv[1]); }
  int result = our_code_starts_here(input, MEMORY);
  print_val(result);
  printf("\n");
  fflush(stdout);
  return 0;
}
```

```
union snake_val {
  int as_int;
  union snake_val* as_ptr;
};
extern union snake_val our_code_starts_here()
  asm("our_code_starts_here");

void print_val(union snake_val val) {
  if(val.as_int & 1) {
    printf("%d", (val.as_int - 1) / 2);
  }
  else if (val.as_int == 6) { printf("true"); }
  else if (val.as_int == 2) { printf("false"); }
  else { // It's a pair!









  }
}
int main(int argc, char** argv) {
  int input = 0;
  int* MEMORY = calloc(10000, sizeof(int));
  if(argc > 1) { input = atoi(argv[1]); }
  union snake_val result;
  result = our_code_starts_here(input, MEMORY);
  print_val(result);
  printf("\n");
  fflush(stdout);
  return 0;
}
```

```
extern int our_code_starts_here()
  asm("our_code_starts_here");

void print_val(int val) {
  if(val & 1) { printf("%d", (val - 1) / 2); }
  else if (val == 6) { printf("true"); }
  else if (val == 2) { printf("false"); }
  else {









  }
}

int main(int argc, char** argv) {
  int input = 0;

  int* MEMORY = calloc(10000, sizeof(int));

  if(argc > 1) { input = atoi(argv[1]); }
  int result = our_code_starts_here(input, MEMORY);
  print_val(result);
  printf("\n");
  fflush(stdout);
  return 0;
}
```

```
union snake_val {
  int as_int;
  union snake_val* as_ptr;
};
extern union snake_val our_code_starts_here()
  asm("our_code_starts_here");

void print_val(union snake_val val) {
  if(val.as_int & 1) {
    printf("%d", (val.as_int - 1) / 2);
  }
  else if (val.as_int == 6) { printf("true"); }
  else if (val.as_int == 2) { printf("false"); }
  else { // It's a pair!









  }
}
int main(int argc, char** argv) {
  int input = 0;
  int* MEMORY = calloc(10000, sizeof(int));
  if(argc > 1) { input = atoi(argv[1]); }
  union snake_val result;
  result = our_code_starts_here(input, MEMORY);
  print_val(result);
  printf("\n");
  fflush(stdout);
  return 0;
}
```

Value	Representation (bits)	Representation hex	Representation decimal	Useful C type
	31-bit, 2's complement number			
9	0000 0000 0000 0000 0000 0000 0001 **0011**	0x00000013	19	int
-2	1111 1111 1111 1111 1111 1111 1111 **1101**	0xFFFFFFFD	-3	int
true	0000 0000 0000 0000 0000 0000 0000 **0110**	0x00000006	6	int
false	0000 0000 0000 0000 0000 0000 0000 **0010**	0x00000002	2	int
(pair 1 2)	XXXX XXXX XXXX XXXX XXXX XXXX XXXX XX**00**	0xXXXXXXXX	big num	_____

tag bits (X1=num, 10=bool, 00=pair)