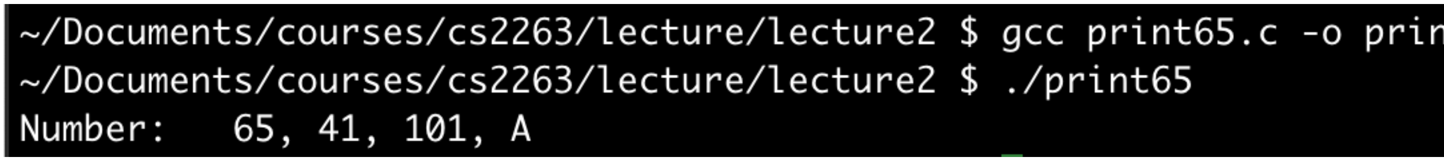
ForNextDay(Lecture2)

Stephen Cole
3553803
CS2263

print65.c

```
#include<stdlib.h>
#include<stdio.h>
int main()
{
	int val = 65;
	printf("Number: %4d, %x, %o, %c \n", val, val, val, (char)val);
	return EXIT_SUCCESS;
}
```



```
~/Documents/courses/cs2263/lecture/lecture2 $ gcc print65.c -o prin
~/Documents/courses/cs2263/lecture/lecture2 $ ./print65
Number:   65, 41, 101, A
```

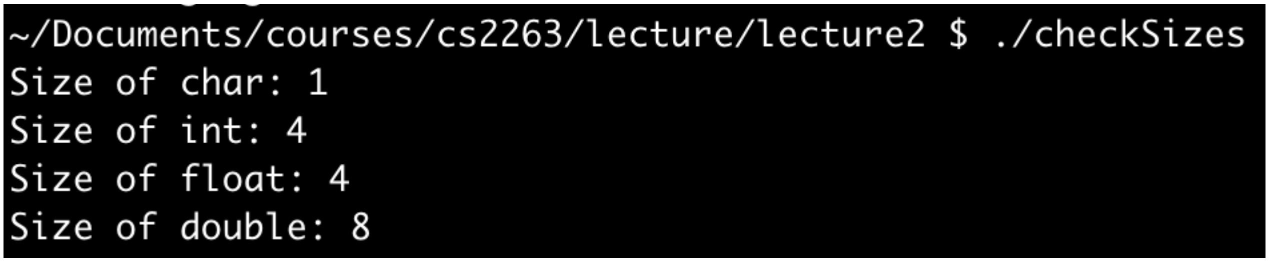
---

checkSizes.c

```
#include<stdlib.h>
#include<stdio.h>
int main()
{
	char exChar;
	int exInt;
	float exFloat;
	double exDouble;

	printf("Size of char: %d \n", sizeof(exChar));
	printf("Size of int: %d \n", sizeof(exInt));
	printf("Size of float: %d \n", sizeof(exFloat));
	printf("Size of double: %d \n", sizeof(exDouble));

	return EXIT_SUCCESS;
}
```



```
~/Documents/courses/cs2263/lecture/lecture2 $ ./checkSizes
Size of char: 1
Size of int: 4
Size of float: 4
Size of double: 8
```

These numbers represent the number of bytes needed to store the data type. A char is stored in 8 bits or 1 byte. Both integers and floats are stored in 32 bits or 4 bytes and a double is stored 64 bits or 8 bytes.

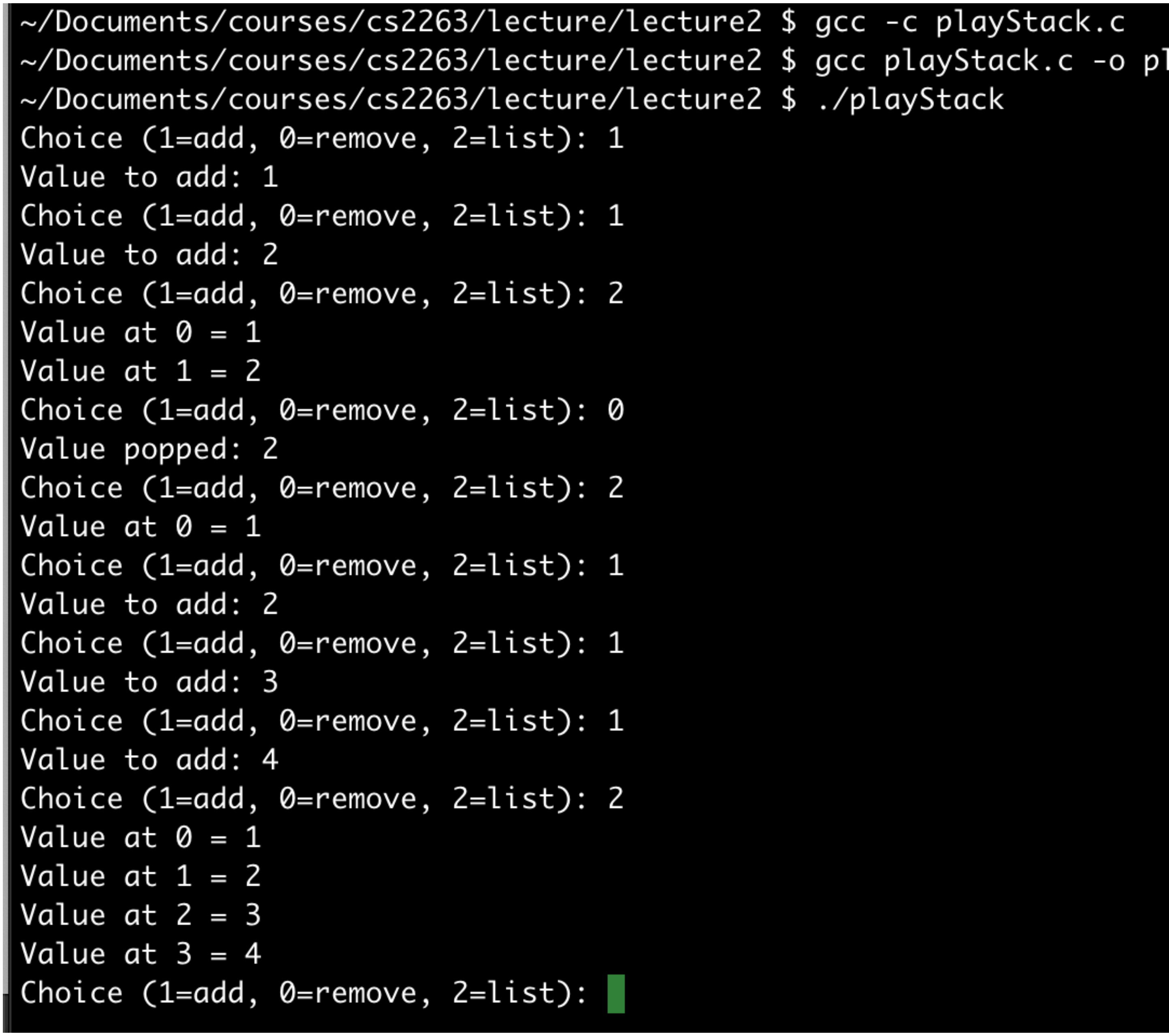
Signed Integer max: 2147483647, min: -2147483647-1

---

playStack.c

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 256
#define PUSH 1
#define POP 0
#define LIST 2
int main(int argc, char* argv[])
{
  int stack[MAX];
  int size = 0;
  int val;
  int iChoice;
  int iNRead;

  /* Processing loop */
  printf("Choice (1=add, 0=remove, 2=list): ");
  iNRead = scanf("%d", &iChoice);
  while(iNRead == 1)
  {
    switch(iChoice)
    {
      case PUSH:
        printf("Value to add: ");
                                        scanf("%d", &val);
                                        if(size < MAX) {
                                                  stack[size] = val;
                                                  size++;
                                        }
    // Read the element, add it to the stack
   break;
   case POP:
                        if(size > 0){
                              size--;
                              val = stack[size];
                              printf("Value popped: %d \n", val);
                        }
   // Print out the last element and remove it.
   break;
   case LIST:
                        for(int i=0; i<size; i++)
                        {
                              printf("Value at %d = %d \n", i, stack[i]);
                        }
    // Print out the stack elements
   break;
  }
  printf("Choice (1=add, 0=remove, 2=list): ");
  iNRead = scanf("%d", &iChoice);
 }
 return EXIT_SUCCESS;
}
```



```
~/Documents/courses/cs2263/lecture/lecture2 $ gcc -c playStack.c
~/Documents/courses/cs2263/lecture/lecture2 $ gcc playStack.c -o pl
~/Documents/courses/cs2263/lecture/lecture2 $ ./playStack
Choice (1=add, 0=remove, 2=list): 1
Value to add: 1
Choice (1=add, 0=remove, 2=list): 1
Value to add: 2
Choice (1=add, 0=remove, 2=list): 2
Value at 0 = 1
Value at 1 = 2
Choice (1=add, 0=remove, 2=list): 0
Value popped: 2
Choice (1=add, 0=remove, 2=list): 2
Value at 0 = 1
Choice (1=add, 0=remove, 2=list): 1
Value to add: 2
Choice (1=add, 0=remove, 2=list): 1
Value to add: 3
Choice (1=add, 0=remove, 2=list): 1
Value to add: 4
Choice (1=add, 0=remove, 2=list): 2
Value at 0 = 1
Value at 1 = 2
Value at 2 = 3
Value at 3 = 4
Choice (1=add, 0=remove, 2=list): 
```