

1. Given the following complete C program, give the portion of memory that each variable is stored in (1 point each).

```
#include <stdio.h>

extern int y;

void somefunction()
{
    static int x = 11;
}

char s[] = "hello";

int main(int argc, char *argv[])
{
    int a = 3;
    int b[] = {1, 2, 3};
    int *c = (int *)malloc(sizeof(int) * 5);
    int **d = (int **)malloc(sizeof(int *));
    d[0] = b;
    return 0;
}
```

- (a) x
 - (b) argv[2]
 - (c) b
 - (d) b[0]
 - (e) a
 - (f) s
 - (g) c[2]
 - (h) c
 - (i) y
 - (j) (*d)[0]
2. Assume we have the following bitwise operations defined:

```
unsigned char set(unsigned char x, int i);
unsigned char clear(unsigned char x, int i);
unsigned char get(unsigned char x, int i);
unsigned char toggle(unsigned char x, int i);
```

Evaluate the following operations by hand (SHOW YOUR WORK)(2 points each):

- (a) get(11, 2)

(b) `set(1, 3)`

(c) `clear(23, 3)`

(d) `toggle(23, 3)`

(e) `get(set(45, 4) » 3, 1)`

3. For each of the following, give the correct portion of C code (2 points each):

(a) Give the portion of C code that sums all of the elements in the following array into a variable called `sum`.

```
int a[10];
```

- (b) Give the line of C code that allocates the proper amount of memory to store 5 characters to a variable called `s`.

 - (c) Give the portion of C code that checks if the char type variable `c` is a digit or a lowercase letter or neither. Print "digit", "lowercase", or "neither".
4. Give a complete C program for one of the following prompts (4 points):
- (a) Write a complete C program that continuously reads integers into an array, growing the size of the array as needed.
 - (b) Write a complete C program that reads an unsigned int as input. Count the number of 0 bits in its binary representation, which we will call n . Output 2^n .