CS232 Computer Organization Spring 2021

Assignment#3 Part 1, due end of Saturday, 02/20/2020

Question 1

Given the following C code snippet, list the VALUE and TYPE of each expression below.

```
int x = 3, i, myarray[10];
float f = 3.4;
for(i=0; i < 10; i++) {
   myarray[i]  = i % x;
}
```

i	0	1	2	3	4
a[i]	0	0	0	0	1

1. x + f

value: 6.4
type: float

2. myarray[4]

value: 1
type: int

3. myarray[4] > myarray[3]

value: 1
type: int

4. myarray

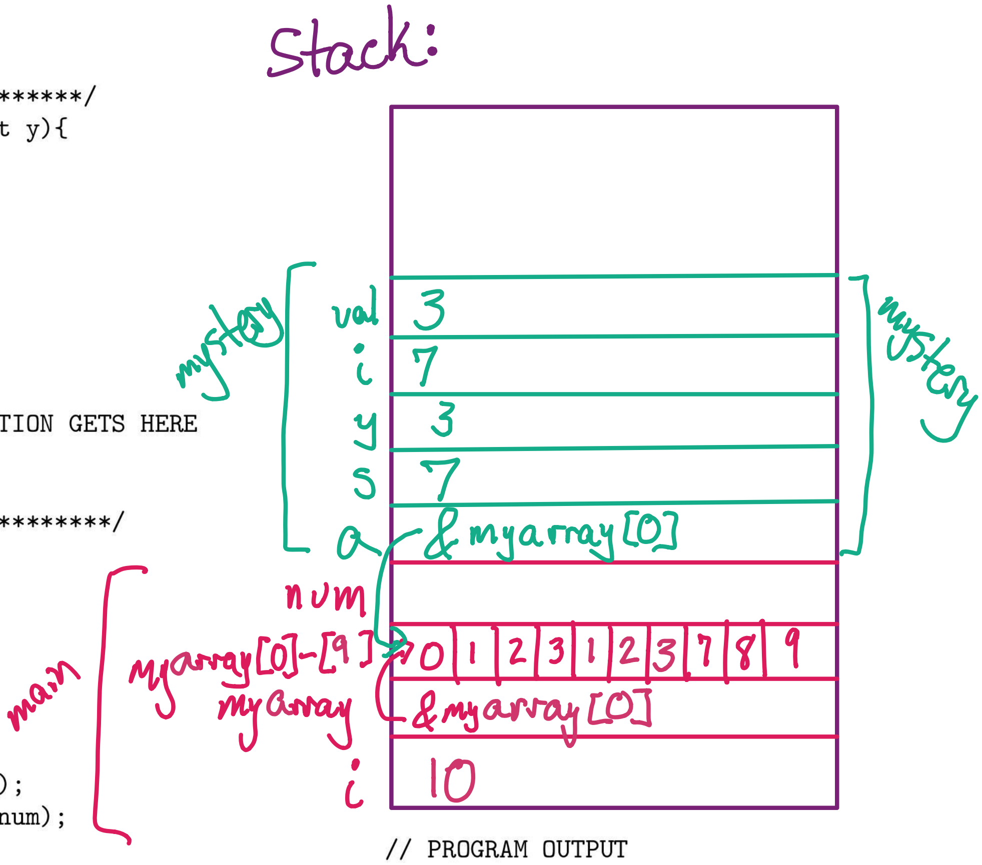
value: &myarray[0]
type: pointer

Question 2

Trace through the following C code, and draw the stack at the execution point indicated in mystery, and show the output produced by a complete run of the program. (Assume stdio.h has been included.)

```
void print_array(int a[], int s) {
  int i;
  for(i=0; i < s; i++) {
    printf("%d:%d, ", i, a[i]);
  }
  printf("\n");
}
/************************************/
int mystery(int a[], int s, int y){
  int i, val;
  val = 0;
  for(i = 0; i < s; i++) {
      if(a[i] > y) {
        val++;
        a[i] = a[i] - y;
      }
  }
  // DRAW THE STACK WHEN EXECUTION GETS HERE
  return val;
}
/**************************************/
int main() {
  int i, myarray[10], num;
  for(i=0; i < 10; i++) {
     myarray[i] = i;
  }
  printf("Before:\n");
  print_array(myarray, 10);
  num =  mystery(myarray, 7, 3);
  printf("After: num = %d\n", num);
  print_array(myarray, 10);
}
```

// YOUR STACK DRAWING



// PROGRAM OUTPUT

Before:
0:0, 1:1, 2:2, 3:3, 4:4, 5:5, 6:6, 7:7, 8:8, 9:9,
After: num = 3
0:0, 1:1, 2:2, 3:3, 4:1, 5:2, 6:3, 7:7, 8:8, 9:9,

Question 3

Consider the following declarations and assignments:

```
int *a, b[5], c, *d;

for (c=0; c < 5 ; c++)  {
    b[c]= 1+c;
}
d=b;
a = &c;
c = d[3];
```

What are the TYPE and VALUE of each of the following expressions (if the expression is invalid, write “Illegal Expression”, and if it is an address describe what it is the address of):

TYPE VALUE

1. a pointer to int &c (address of c)

2. b pointer to int &b[0] (address of b[0])

3. c int 4

4. &b[1] address address of b[1]

5. d pointer to int &b[0] (address of b[0])

6. *d int 1

Question 4

Trace through the following C code, and draw memory contents (heap and stack) at the execution point indicated in foo, and show the output produced by a complete run of the program. (Assume stdio.h and stdlib.h have been included, and that malloc succeeds.)

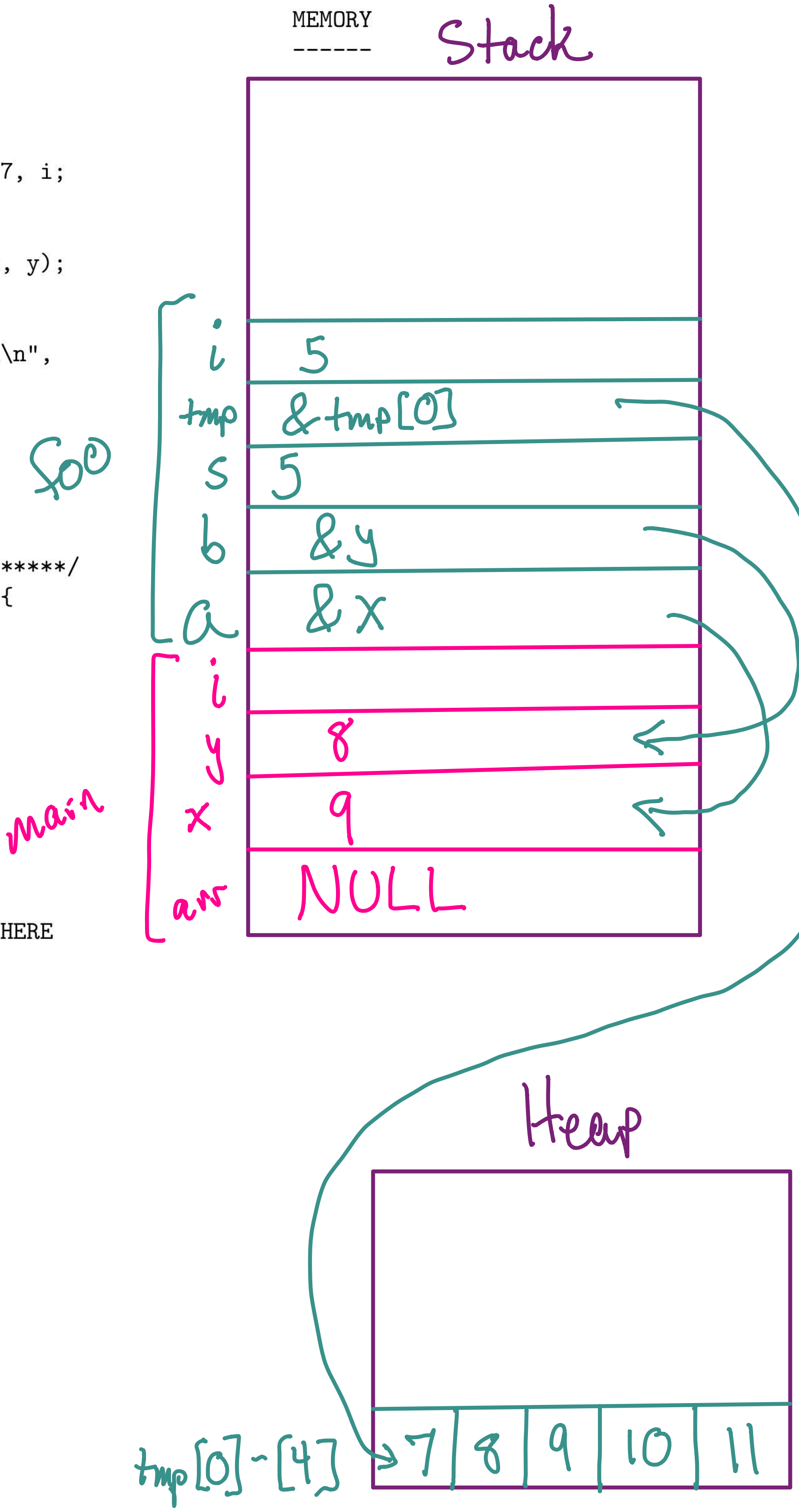
```
int *foo(int *a, int *b, int s);

int main () {
    int *arr = NULL, x = 6, y = 7, i;

    arr = foo(&x, &y, 5);
    printf("x = %d  y = %d\n", x, y);
    if(arr != NULL) {
        for(i=0; i < 5; i++) {
            printf("arr[%d] = %d\n",
                    i, arr[i]);
        }
    }
    free(arr);
    return 0;
}
/************************************/
int *foo(int *a, int *b, int s) {
    int *tmp, i;

    tmp = malloc(sizeof(int)*s);
    if(tmp != NULL) {
        for(i=0; i < s; i++) {
            tmp[i] = i + *b;
        }
        *a = tmp[2];
        *b = 8;
    }
    // DRAW MEMORY WHEN YOU GET HERE
    return tmp;
}
```

MEMORY



OUTPUT

```
x=9   y=8
arr[0]=7
arr[1]=8
arr[2]=9
arr[3]=10
arr[4]=11
```