William Poole

Pascal Francis-Mezger

ECE 177

16 April 2021

Question 1:

What will y be at the end of the program?

```
int x[]={8,6,12,4,2,7};
int *ptr;
int y;
ptr=x;
y=*(ptr+3);
```

ptr memory is at the first index of the array and when you add three you arrive at the fourth.

Y=4 at the end of the program

Question 2:

Values of y and z at the end of the program?

```
1 int x[]={8,6,12,4,2,7};
2 int *ptr;
3 int y;
4 int z;
5 ptr=x;
6 y=*(ptr++);
7 z=*(++ptr);
```

y is equal to 8 because the increment occurs after the access, z equals 12 due to it being affected by two increments putting at 12 in the array.

Y=8

Z=12

Question 3:

Malloc vs. calloc, why not always use calloc?

Malloc generates an area in memory of a certain size, depending on input, and returns the first position as a pointer. (does not initialize memory).

Calloc generates an area in memory of a certain size, depending on input, and returns the first position of the area as a pointer. (initializes memory, set to 0).

Calloc could take to long and it can erase values that maybe in those areas, where malloc would just exist over the area.

Question 4:

Code

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int amount=300;
    float *value = calloc(amount,sizeof(float));
    printf("%f",*value);

    return 0;
}
```

```
input
0.000000

...Program finished with exit code 0
Press ENTER to exit console.
```