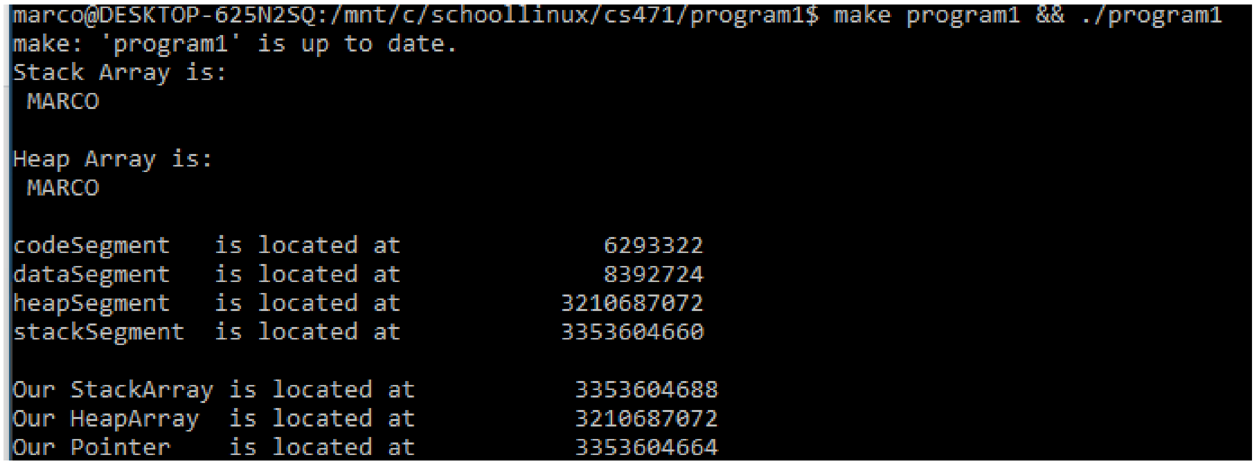
Question Answers:

1) code (also a file will be included):

```
/*
Name: Marco Salazar
Date: 8/26/2020
Assignment: Programming assignment 1

There is no input in this program. The only output will be words printed out
using a int array buffer.
there are no post or pre conditions.
*/
#include <stdio.h>
#include <malloc.h>

int dataSegment;

int main() {
        int stackSegment;
        int *heapSegment;
        heapSegment = malloc(100);

        int A[20];
        A[0] = 'M' + 'A'*256 + 'R'*256*256 + 'C'*256*256*256;
        A[1] = 'O'          + '!'*256*256 + '!'*256*256*256;
        A[2] = 0;
        heapSegment[0] = 'M' + 'A'*256 + 'R'*256*256 + 'C'*256*256*256;
        heapSegment[1] = 'O';
        heapSegment[2] = 0;


        char *S = (char *) A;
        printf("Stack Array is:\n %s\n\n", S);
        char *H = (char *) heapSegment;
        printf("Heap Array is:\n %s\n\n", H);

        printf("codeSegment   is located at %20u\n", main);
        printf("dataSegment   is located at %20u\n", &dataSegment);
        printf("heapSegment   is located at %20u\n", heapSegment);
        printf("stackSegment  is located at %20u\n", &stackSegment);
        printf("\n");
        printf("Our StackArray is located at %20u\n", A);
        printf("Our HeapArray  is located at %20u\n", heapSegment);
        printf("Our Pointer    is located at %20u\n", &S);
        return 0;
    }
```

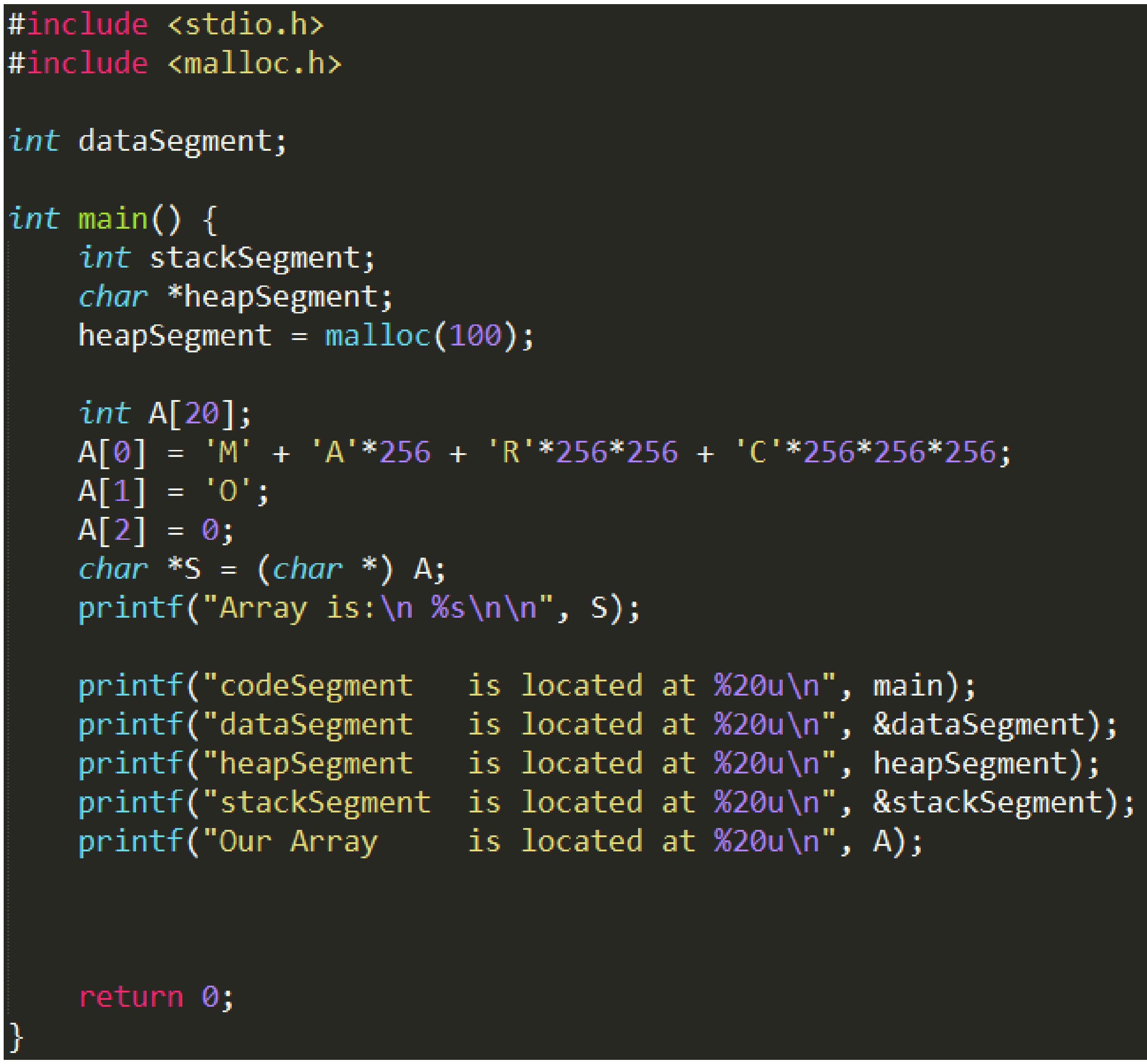
2) Screenshot:



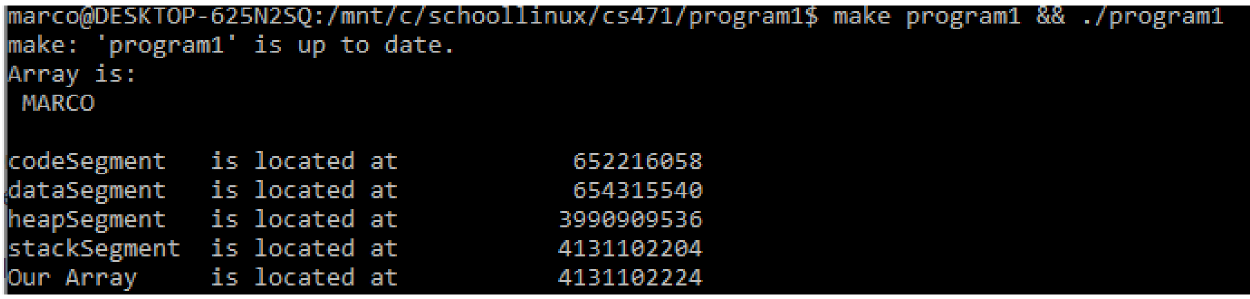
```
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs471/program1$ make program1 && ./program1
make: 'program1' is up to date.
Stack Array is:
 MARCO

Heap Array is:
 MARCO

codeSegment   is located at              6293322
dataSegment   is located at              8392724
heapSegment   is located at           3210687072
stackSegment  is located at           3353604660

Our StackArray is located at           3353604688
Our HeapArray  is located at           3210687072
Our Pointer    is located at           3353604664
```

3) Question Answers:

a. The Memory segment that the array is allocated in is in the stack segment:

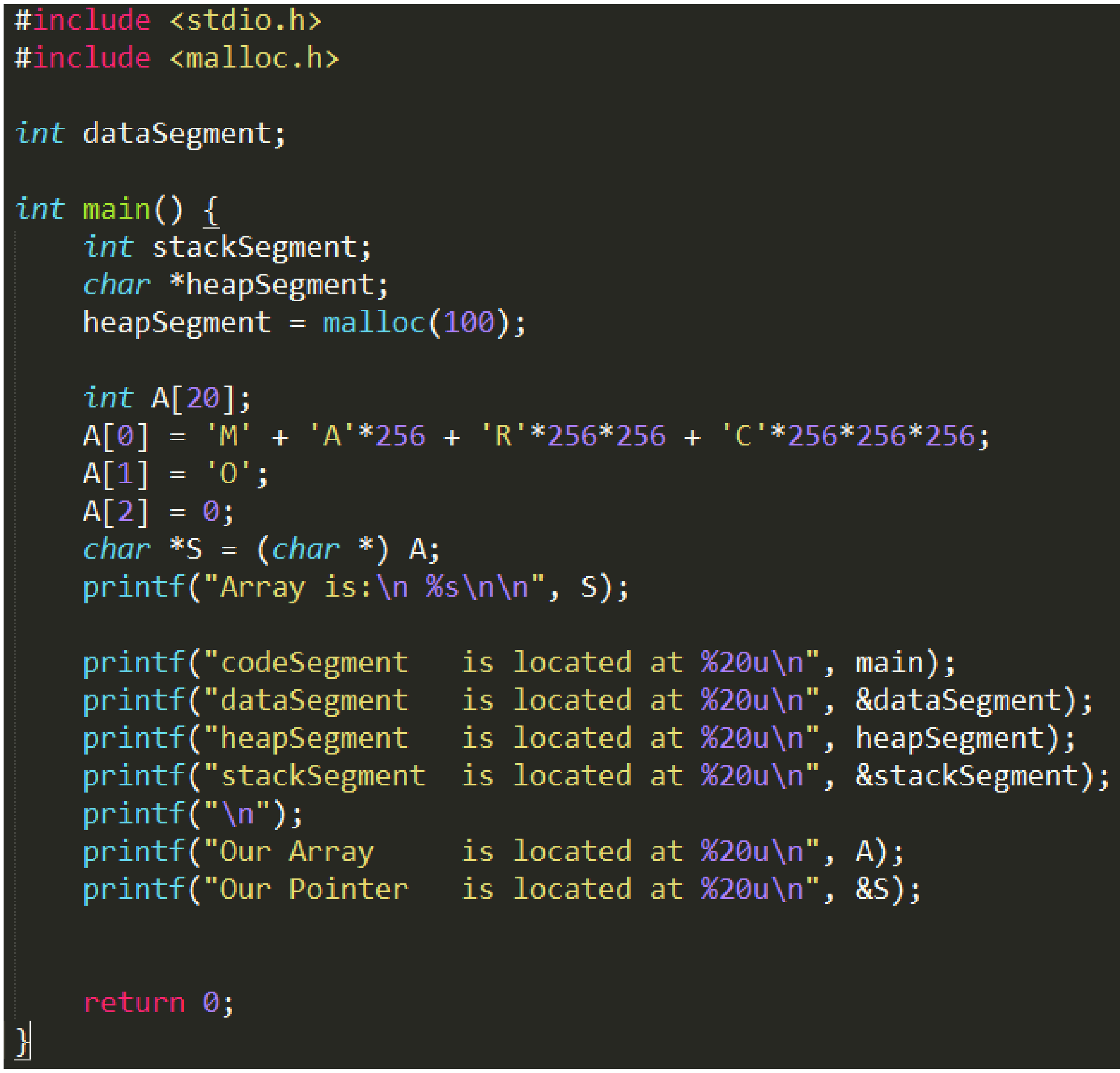


```c
#include <stdio.h>
#include <malloc.h>

int dataSegment;

int main() {
    int stackSegment;
    char *heapSegment;
    heapSegment = malloc(100);

    int A[20];
    A[0] = 'M' + 'A'*256 + 'R'*256*256 + 'C'*256*256*256;
    A[1] = 'O';
    A[2] = 0;
    char *S = (char *) A;
    printf("Array is:\n %s\n\n", S);

    printf("codeSegment   is located at %20u\n", main);
    printf("dataSegment   is located at %20u\n", &dataSegment);
    printf("heapSegment   is located at %20u\n", heapSegment);
    printf("stackSegment  is located at %20u\n", &stackSegment);
    printf("Our Array     is located at %20u\n", A);



    return 0;
}
```

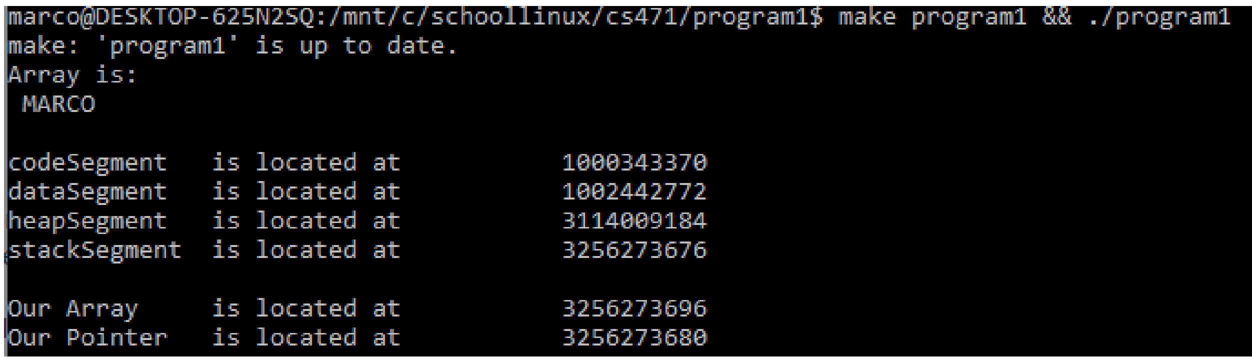


```
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs471/program1$ make program1 && ./program1
make: 'program1' is up to date.
Array is:
 MARCO

codeSegment   is located at            652216058
dataSegment   is located at            654315540
heapSegment   is located at           3990909536
stackSegment  is located at           4131102204
Our Array     is located at           4131102224
```

b. The pointer is located in the stack segment:

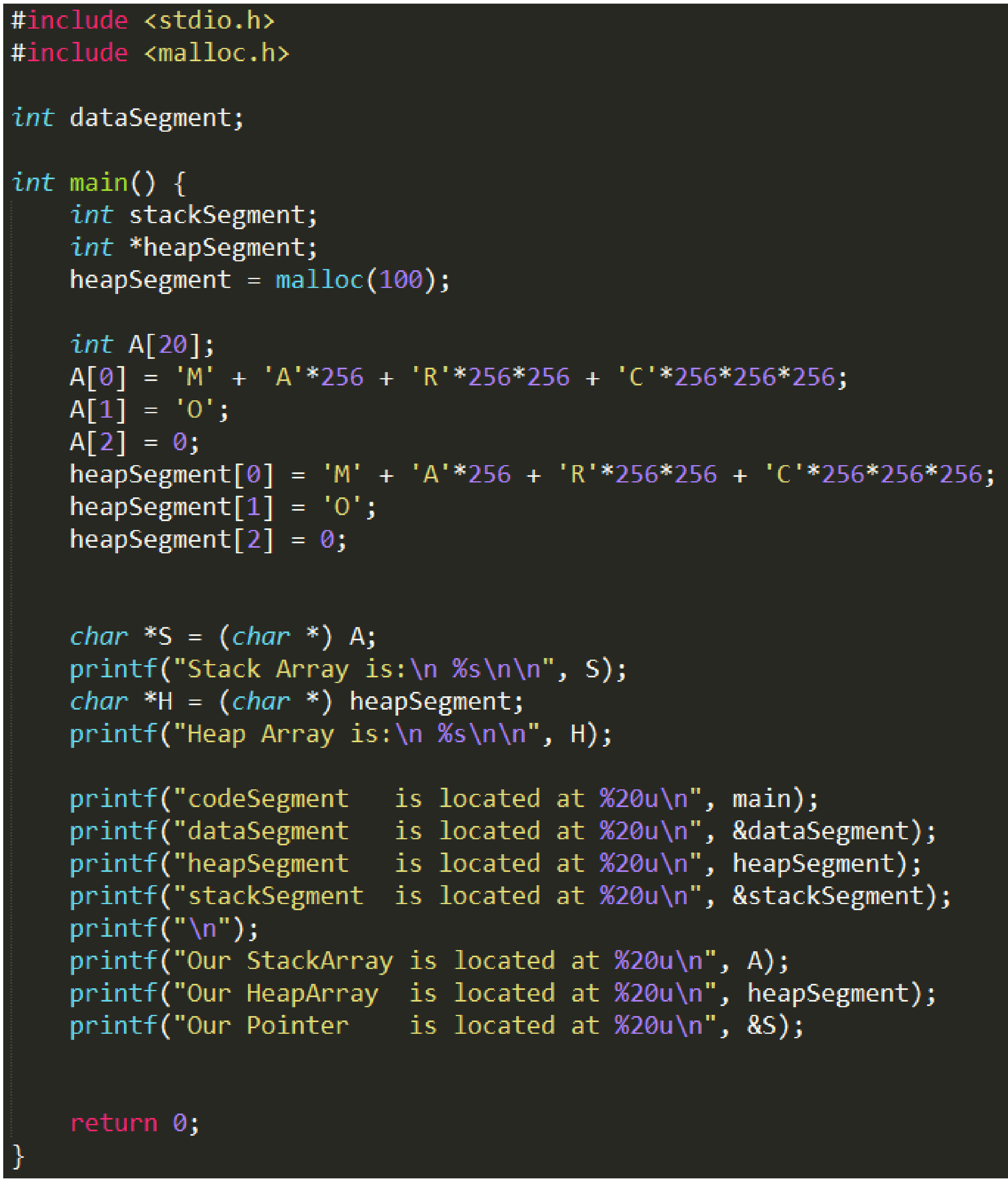


```c
#include <stdio.h>
#include <malloc.h>

int dataSegment;

int main() {
    int stackSegment;
    char *heapSegment;
    heapSegment = malloc(100);

    int A[20];
    A[0] = 'M' + 'A'*256 + 'R'*256*256 + 'C'*256*256*256;
    A[1] = 'O';
    A[2] = 0;
    char *S = (char *) A;
    printf("Array is:\n %s\n\n", S);

    printf("codeSegment   is located at %20u\n", main);
    printf("dataSegment   is located at %20u\n", &dataSegment);
    printf("heapSegment   is located at %20u\n", heapSegment);
    printf("stackSegment  is located at %20u\n", &stackSegment);
    printf("\n");
    printf("Our Array     is located at %20u\n", A);
    printf("Our Pointer   is located at %20u\n", &S);


    return 0;
}
```

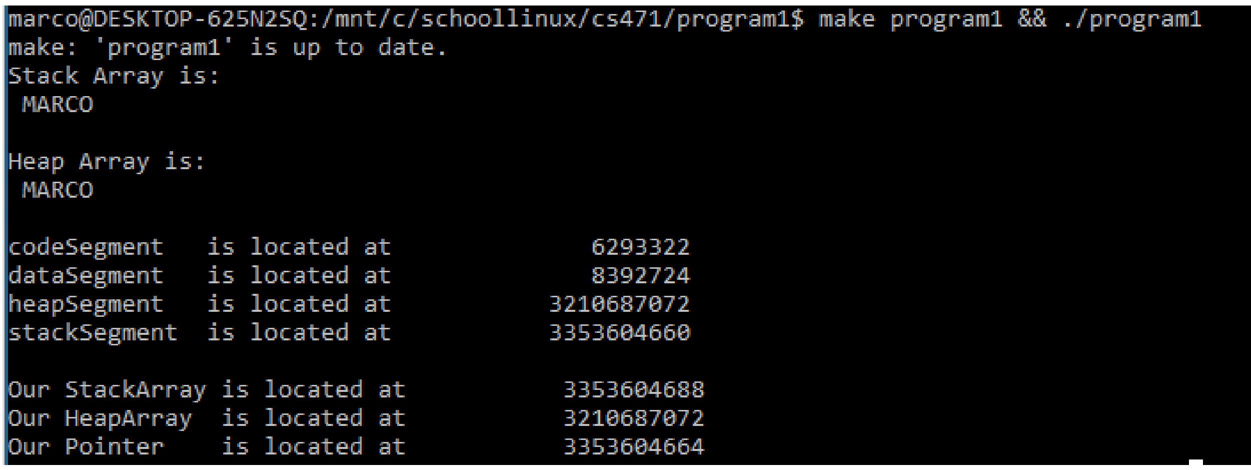


```
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs471/program1$ make program1 && ./program1
make: 'program1' is up to date.
Array is:
 MARCO

codeSegment   is located at           1000343370
dataSegment   is located at           1002442772
heapSegment   is located at           3114009184
stackSegment  is located at           3256273676

Our Array     is located at           3256273696
Our Pointer   is located at           3256273680
```

c. One way is you can put it in the heap:



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

int dataSegment;

int main() {
    int stackSegment;
    int *heapSegment;
    heapSegment = malloc(100);

    int A[20];
    A[0] = 'M' + 'A'*256 + 'R'*256*256 + 'C'*256*256*256;
    A[1] = 'O';
    A[2] = 0;
    heapSegment[0] = 'M' + 'A'*256 + 'R'*256*256 + 'C'*256*256*256;
    heapSegment[1] = 'O';
    heapSegment[2] = 0;


    char *S = (char *) A;
    printf("Stack Array is:\n %s\n\n", S);
    char *H = (char *) heapSegment;
    printf("Heap Array is:\n %s\n\n", H);

    printf("codeSegment   is located at %20u\n", main);
    printf("dataSegment   is located at %20u\n", &dataSegment);
    printf("heapSegment   is located at %20u\n", heapSegment);
    printf("stackSegment  is located at %20u\n", &stackSegment);
    printf("\n");
    printf("Our StackArray is located at %20u\n", A);
    printf("Our HeapArray  is located at %20u\n", heapSegment);
    printf("Our Pointer    is located at %20u\n", &S);


    return 0;
}
```

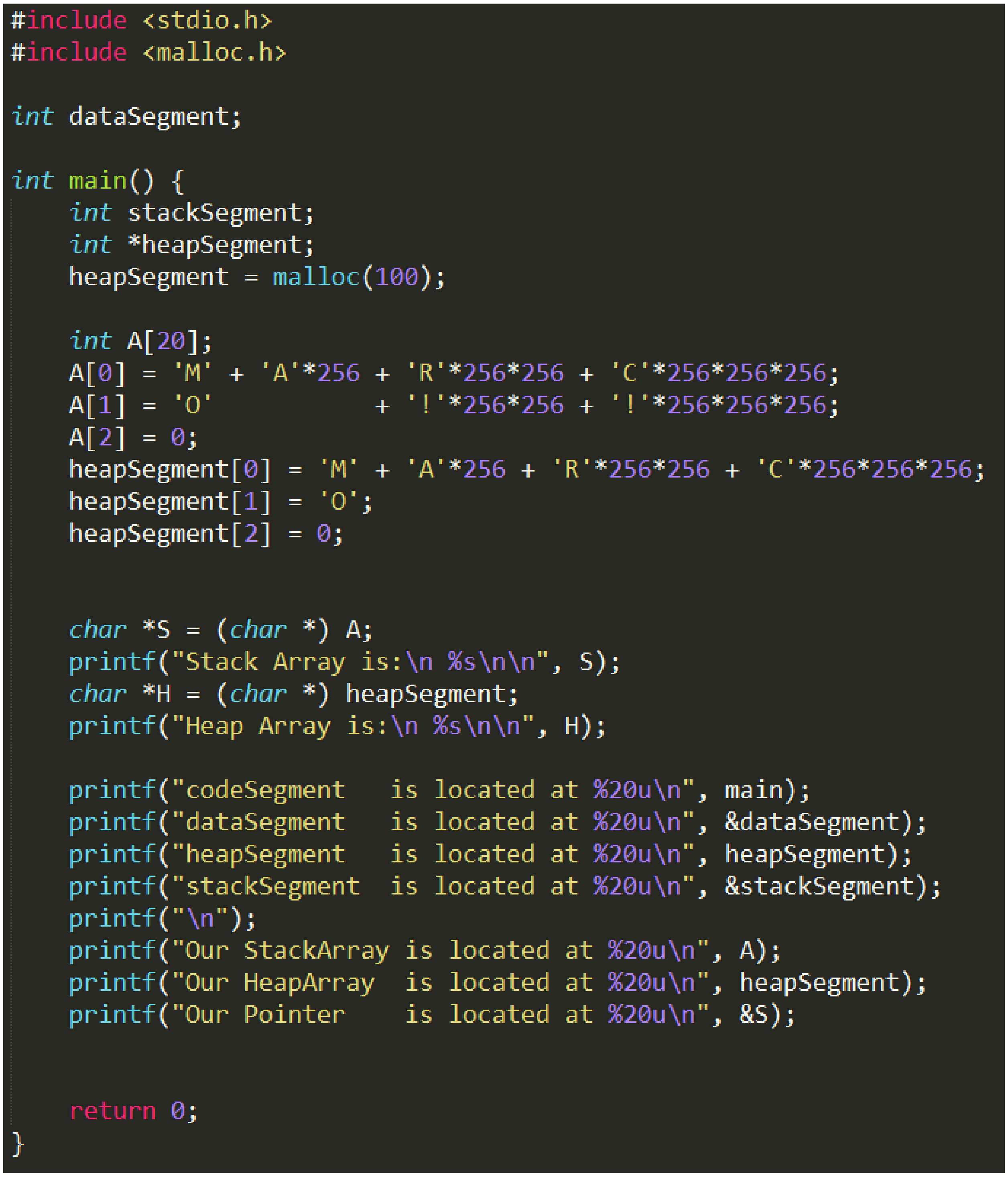


```
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs471/program1$ make program1 && ./program1
make: 'program1' is up to date.
Stack Array is:
 MARCO

Heap Array is:
 MARCO

codeSegment   is located at              6293322
dataSegment   is located at              8392724
heapSegment   is located at           3210687072
stackSegment  is located at           3353604660

Our StackArray is located at           3353604688
Our HeapArray  is located at           3210687072
Our Pointer    is located at           3353604664
```

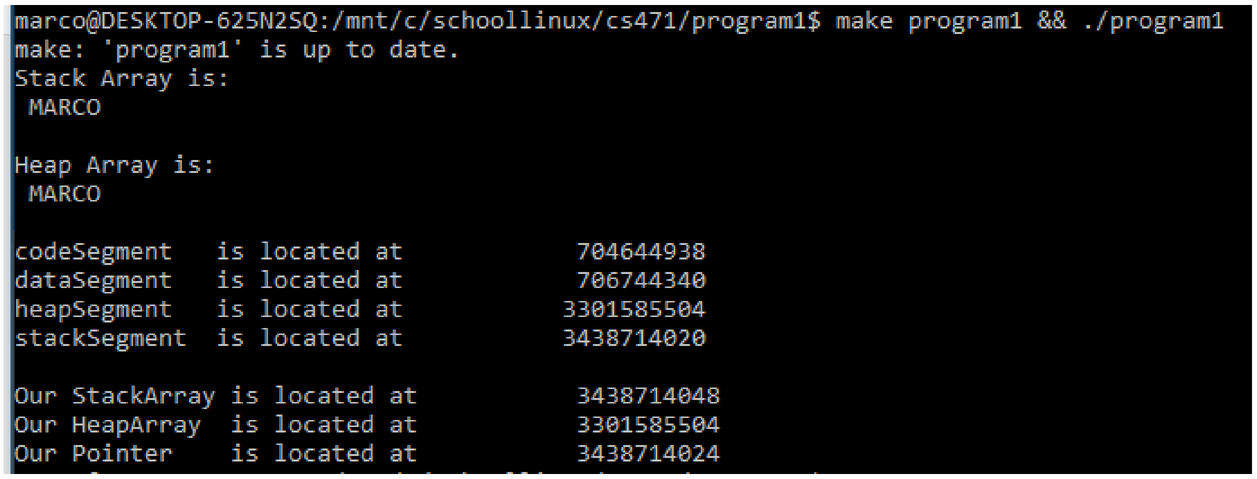
d. My computer is Little Endian.

e. It seems that there are a couple different philosophies that led to big endian and little endian. Many sources say that one of the reasons little endian is good, is that "the address of a given value in memory, taken as a 32, 16, or 8 bit width, is the same." (source). It also allows for more efficiency in addition and subtraction in older systems. On the other hand, Big Endian makes it very easy to tell whether a number is positive or negative, as well as estimating its size (source). Altogether though, I believe it would depend on the context as to which one is better. I would personally think that since the differences are so minor, it would be best to use the one that most applications and OS's like to deal with like Little Endian.

4) We can just fill the last byte with 0. I proved that by putting exclamation points between the two options such that, if filling the last byte is sufficient, the exclamation points will not be printed.



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

int dataSegment;

int main() {
    int stackSegment;
    int *heapSegment;
    heapSegment = malloc(100);

    int A[20];
    A[0] = 'M' + 'A'*256 + 'R'*256*256 + 'C'*256*256*256;
    A[1] = 'O'           + '!'*256*256 + '!'*256*256*256;
    A[2] = 0;
    heapSegment[0] = 'M' + 'A'*256 + 'R'*256*256 + 'C'*256*256*256;
    heapSegment[1] = 'O';
    heapSegment[2] = 0;


    char *S = (char *) A;
    printf("Stack Array is:\n %s\n\n", S);
    char *H = (char *) heapSegment;
    printf("Heap Array is:\n %s\n\n", H);

    printf("codeSegment   is located at %20u\n", main);
    printf("dataSegment   is located at %20u\n", &dataSegment);
    printf("heapSegment   is located at %20u\n", heapSegment);
    printf("stackSegment  is located at %20u\n", &stackSegment);
    printf("\n");
    printf("Our StackArray is located at %20u\n", A);
    printf("Our HeapArray  is located at %20u\n", heapSegment);
    printf("Our Pointer    is located at %20u\n", &S);


    return 0;
}
```



```
marco@DESKTOP-625N2SQ:/mnt/c/schoollinux/cs471/program1$ make program1 && ./program1
make: 'program1' is up to date.
Stack Array is:
 MARCO

Heap Array is:
 MARCO

codeSegment   is located at             704644938
dataSegment   is located at             706744340
heapSegment   is located at            3301585504
stackSegment  is located at            3438714020

Our StackArray is located at            3438714048
Our HeapArray  is located at            3301585504
Our Pointer    is located at            3438714024
```