BBM 201

Exercise Set 1

1. Assume the variable declarations given below:
   ```
   int a = 2;
   int* b= &a;
   ```
   Which of the following statement(s) will change the value of a to 1?

   (a) b--;

   (b) a--;

   (c) (*a)--;

   (d) (*b)--;

2. Which code fragment(s) could be inserted in the blank in order to safely initialize each element of A to zero?
   ```
   int* p=&A[0];
   for(int i=0; i<size; i++,p++) .........
   ```
   (a) *A=0;

   (b) A[i]=0;

   (c) *p=0;

   (d) *i=0;

3. What is the output of the following code fragment?

   ```
   string s;
   int* i;
   s="Fred Cakmaktas";
   i=&s;
   printf("*i");
   ```

4. What is the output of the following code?
   ```
   void aMethod(int* p){
       (*p)=(*p)*3;
   }
   int main(){
       int a=2;
       aMethod(&a);
       printf("a");
   }
   ```

5. (a) What is the output of the following program?

(b) Please change the program so that it returns the sum of the elements of the array
A as the variable total.

```c
#include<stdio.h>
int SumOfElements(int A[])
{
    int i, sum = 0;
    int size = sizeof(A)/sizeof(A[0]);
    printf("SOE - Size of A = %d, size of A[0] = %d",sizeof(A),sizeof(A[0]));
    for(i = 0;i< size;i++)
    {
        sum+= A[i];
    }
    return sum;
}
int main()
{
    int A[]= {1,2,3,4,5};
    int total = SumOfElements(A);
    printf("Sum of elements = %d\n",total);
    printf("Main - Size of A = %d, size of A[0] = %d",sizeof(A),sizeof(A[0]));
```

6. The following are declarations of arrays in C. Circle whether each lives on the stack or on the heap.
   a) int x[5];

   Stack   Heap

   b) int *p;
   p = (int*)malloc(sizeof(int));

   Stack   Heap

   c) int a[2][4];

   Stack   Heap

7. The following are declarations of arrays in the heap in C. Write the code that frees their memory.

   a) int *p;
       p = (int*)malloc(sizeof(int));
       *p = 10;

   b) void *p = malloc(3*sizeof(int));

8. Write a function called swap() that can be used to swap integers that, when used in the following program, would produce the expected output of "3, 4". Fill in the blank for

the arguments to your new function.

```
 void swap(...)                  #include<stdio.h>
 {                               int main ()
                                 {
                                     int x = 4;
                                     int y = 3;
                                 swap(...);
                                 printf(''%d, %d'', x, y);
 }                               }
```