1. [6 points] What will be the output of the following program? Explain how you reached your solution.

```c
#include<stdio.h>

int main()
{
    int a[5] = {5, 1, 15, 20, 25};
    int x, y, z;
    x = ++a[1];
    y = a[1]++;
    z = a[x++];
    printf("%d, %d, %d", x, y, z);
    return 0;
}
```

2. [2 points] In C, you have to free the memory you allocated. In language like Java, the garbage collector is doing this for you. State one advantage for each strategy.

3. [4 points] Write a C function, called `compare()`, (no need to write a full-fledged program, just the function) that takes two integers as arguments and returns 1 if the two numbers are equal and 0 if they are not. However, you must not use any comparison operators.

4. [3 points, 1 for each] Given the unsigned int x, using only shifting left and right, add, and subtract, write the C statements (just statements not full-fledged functions) to calculate the following (each problem below is independent from the others). Try to optimize as much as possible.
   a) $y = 7*x$;
   b) $y = 27*x$;
   c) $y = 67*x$;
5. \[
\begin{align*}
\text{int } & \text{array1}[M][N]; \\
\text{int } & \text{array2}[N][M]; \\
\text{int } & \text{copy(int } i, \text{ int } j) \\
& \{ \\
& \quad \text{array1}[i][j] = \text{array2}[j][i]; \\
& \} \\
\end{align*}
\]

Suppose the above code generates the following assembly code (assume array2 and array1 are the base addresses of the corresponding arrays and assume that M and N in the above code has been define with \#define at the beginning of the program and not shown here):

```assembly
    copy:
        movl %rdi, %ecx
        movl %rsi, %ebx
        leal (%ecx,%ecx,8), %edx
        sall $2, %edx
        movl %ebx, %eax
        sall $4, %eax
        subl %ebx, %eax
        sall $2, %eax
        movl array2(%eax,%ecx,4), %eax
        movl %eax, array1(%edx,%ebx,4)
        ret
```

What are the values of M and N (4 points)?
Show how did you reach your answer (6 points)
6. Consider the two C functions fun1 and fun2. Which of these two functions compiled into the assembly code shown[1 point]? How did you reach this conclusion [4 points]?

```c
int fun1(int a, int b)
{
    if (a < b)
        return a;
    else
        return b;
}

int fun2(int a, int b)
{
    if (b < a)
        return b;
    else
        return a;
}
```

```assembly
movl %rdi, %edx
movl %rsi, %eax
cmpl %eax, %edx
jge .L9
movl %edx, %eax
.L9:
```