Answers to Questions 1-5:

(Czy)

1. Yes, *x multiply by 19* can be implemented by using shifts and adds only. Notice how x * 19 is equivalent to (x * 16) + (x * 2) + x. Subsequently, multiplying by 16 is equivalent to shifting to the left by 4 and multiplying x by 2 is equivalent to shifting to the left by 1. From this, the code implementation can be as follows:

   ```
   int dummy(int x){
     int ret = (x << 4) + (x << 1) + x;
     return ret;
   }
   ```

(Czy)

2. Implementing only shifts and adds modifies the assembly code such that it longer uses multiply instruction. Instead, it only uses logical shift left and additions

(Gab)

3. For the case of x * 45, the C++ code should be modified as follows:

   ```
   int dummy(int x){
     int ret = (x << 5) + (x << 3) + (x << 2) + x;
     return ret;
   }
   ```

(Kenjie)

4. For the case of x * (-2), the C++ code should be modified as follows:

   ```
   int dummy(int x){
     int ret = -(x << 1);
     return ret;
   }
   ```

(Czy)

5. For the case of x * 0, the C++ code should be modified as follows:

   ```
   int dummy(int x){
     return 0;
   }
   ```