Week 04 Tutorial Questions

1. The MIPS processor has 32 general purpose 32-bit registers, referenced as `$0` .. `$31`. Some of these registers are intended to be used in particular ways by programmers and by the system. For each of the registers below, give their symbolic name and describe their intended use:

    a. `$0`
    b. `$1`
    c. `$2`
    d. `$4`
    e. `$8`
    f. `$16`
    g. `$26`
    h. `$29`
    i. `$31`

2. Translate this C program to MIPS assembler

```c
// print the square of a number
#include <stdio.h>

int main(void) {
    int x, y;
    printf("Enter a number: ");
    scanf("%d", &x);
    y = x * x;
    printf("%d\n", y);
    return 0;
}
```

    Store variable **x** in register $t0 and store variable **y** in register **$t1**.

3. Translate this C program so it uses goto rather than if/else.
    Then translate it to MIPS assembler.

```c
#include <stdio.h>

int main(void) {
    int x, y;
    printf("Enter a number: ");
    scanf("%d", &x);

    if (x > 46340) {
        printf("square too big for 32 bits\n");
    } else {
        y = x * x;
        printf("%d\n", y);
    }

    return 0;
}
```

4. Translate this C program so it uses goto rather than if/else.
    Then translate it to MIPS assembler.

```c
#include <stdio.h>

int main(void) {
    int x;
    printf("Enter a number: ");
    scanf("%d", &x);

    if (x > 100 && x < 1000) {
        printf("medium\n");
    } else {
        printf("small/big\n");
    }
}
```

    Consider this alternate version of the above program, use its approach to produce simpler MIPS assembler.

```c
#include <stdio.h>

int main(void) {
    int x;
    printf("Enter a number: ");
    scanf("%d", &x);

    char *message = "small/big\n";
    if (x > 100 && x < 1000) {
        message = "medium";
    }

    printf("%s", message);
}
```

5. Translate this C program so it uses goto rather than if/else.
   Then translate it to MIPS assembler.

```c
#include <stdio.h>

int main(void) {
    for (int x = 24; x < 42; x += 3) {
        printf("%d\n",x);
    }
}
```

6. Translate this C program so it uses goto rather than if/else.
   Then translate it to MIPS assembler.

```c
// print a triangle
#include <stdio.h>

int main (void) {
    for (int i = 1; i <= 10; i++) {
        for (int j = 0; j < i; j++) {
            printf("*");
        }
        printf("\n");
    };
    return 0;
}
```

7. Translate this C program so it uses goto rather than if/else.
   Then translate it to MIPS assembler.

```c
// Simple factorial calculator  - without error checking

#include <stdio.h>

int main (void) {
    int n;
    printf("n  = ");
    scanf("%d", &n);

    int fac = 1;
    for (int i = 1; i <= n; i++) {
        fac *= i;
    }

    printf ("n! = %d\n", fac);
    return 0;
}
```